AFHRL-TR-90-91

DTIC
ELECTE
JAN 27 1992
S B D

(2)

AIR FORCE

AD-A245 573

HUMAN

RESOURCES

METHODOLGY FOR EVALUATION OF AUTOMATION IMPACTS ON TACTICAL COMMAND AND CONTROL (C^2) SYSTEMS: FINAL REPORT

Kevin M. Corker
Nichael L. Cramer

Bolt Beranek and Newman, Incorporated
10 Moulton Street
Cambridge, Massachusetts 02238

Eugene H. Henry, Capt, USAF
Donald E. Smoot, Maj, USAF

LOGISTICS AND HUMAN FACTORS DIVISION
Wright-Patterson Air Force Base, Ohio 45433-6503

December 1990

Final Technical Report for Period January 1990 – November 1990

Approved for public release; distribution is unlimited.

92-01761

LABORATORY

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235-5601

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This report has been reviewed and is approved for publication.

BERTRAM W. CREAM, Technical Director
Logistics and Human Factors Division

JAMES C. CLARK, Colonel, USAF
Chief, Logistics and Human Factors Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
December 1990

3. REPORT TYPE AND DATES COVERED
Final Report – January 1990 – November 1990

4. TITLE AND SUBTITLE
Methodology for Evaluation of Automation Impacts on Tactical Command and Control (C^2) Systems: Final Report

5. FUNDING NUMBERS
C – F33615-87-C-0007
PE – 62205F
PR – 3017
TA – 08
WU – 06

6. AUTHOR(S)
Kevin M. Corker
Nichael L. Cramer
Eugene H. Henry
Donald E. Smoot

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Bolt Beranek and Newman, Incorporated
10 Moulton Street
Cambridge, Massachusetts 02238

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
Logistics and Human Factors Division
Air Force Human Resources Laboratory
Wright-Patterson Air Force Base, Ohio 45433-6503

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFHRL-TR-90-91

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
This effort concerns the research and development of a prototype methodology to assess the impacts of automation on Air Force tactical Command and Control (C^2) systems and operators. The approach uses an object-oriented software paradigm to encode of human operator performance, on-screen C^2 interfaces, and scenarios. Specifically, the prototype methodology provides an environment whereby the analyst can manipulate tactical scenarios and equipment characteristics as they relate to the Control Reporting Center and its automation initiatives. These configurations can then be exercised in conjunction with the human performance models via time-event simulation. Thus, human performance data and operational results are available for examination and, hence, potentially useful as guidance for automation initiatives. This final report describes the introduction of a human-in-loop capability. The methodology originally contained an open-loop emulation of five key C^2 operators and models of their visual, auditory, cognitive, and psychomotor (VACP) respectively. Through voice input and output, as well as touch-panel input, an actual operator can now interact with the emulated operators in a real-time, closed-loop capacity. Thus, human performance data collection and validation of the VACP models are viable.

14. SUBJECT TERMS
automation
C^2 systems
human performance
object-oriented programming
simulation
workload evaluation

15. NUMBER OF PAGES
98

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

SUMMARY

Predicting the impact on performance of introducing automation into dynamic human/machine systems presents a difficult but important challenge to the designer/analyst of such systems. This challenge is even more daunting in the arena of tactical Command and Control (C^2) for advanced Air Force systems, which is the focus of this development effort.

The challenge stems, in part, from the inherent difficulty in representing large-scale dynamic systems with multiple interactive subsystems, and in part from the inclusion of representations for intelligent and autonomous human operators in such systems. There has been a significant shortfall in the development of integrated human performance models that capture the full range of operator behavior characterizing the capabilities and limitations that humans bring to such systems. This difficulty is exacerbated by the time-critical nature of the Command and Control process for ground controlled intercept (GCI) in modern Air Force tactical plans.

The research reported in this document concerns the prototype development of a methodology to assess the impact of automation on command and control (C^2) and battle management systems in the Air Force. This methodology is provided as an early development testbed. It is sufficiently developed to support test and evaluation to determine the direction of continued work needed to move the methodology from a testbed to a turnkey system.

Currently tactical C^2 systems are almost entirely unautomated. There are, however, several ongoing initiatives to automate significant portions of the tactical air control operation in the Air Force. To help guide the implementation of these automation initiatives and to predict the operational impact of this automation, we have developed and implemented a prototype methodology to enable an analyst to simulate human/machine interaction in various automation alternatives. This approach uses an object-oriented software development paradigm to encode human operator behavior, tactical C^2 equipment with varied levels of assumed automation, and tactical scenarios.

In this simulation environment, the analyst can vary operational scenarios and C^2 equipment characteristics in terms of the display/control topology, and then run those

configurations in conjunction with human performance models. Operational results, as well as the output of human performance models, are available for examination.

Effort was taken in the development of this analytic capability to maintain a generality in approach and capability, while bringing the methodology to bear on the automation and manning changes associated with upgrade of the current Control and Reporting Center (CRC) from the 407L configuration to that represented by the Modular Control Equipment (MCE) upgrade as a testbed of the efficacy of the approach.

PREFACE

The design and implementation of an evaluation methodology to assess the impact of automation on the performance of C^2 systems are described in this document. This is the Final Report under USAF Contract #F33615-87-C-0007. The First Interim Report (Methodology for Evaluation of the Impact of Automation on C^2 Systems - AFHRL-TR-89-17) provided information about the context of and requirements for an efficient and effective evaluation methodology to be applied to emerging automation initiatives in the area of tactical C^2. The Second Interim Report detailed the implementation of that evaluation methodology. This document describes the design, development, and software implementation of the evaluation methodology and its operation. The work described is the basis for an ongoing development effort that will include use of the software simulation to investigate human operation of advanced automation in tactical C^2 systems. Provision for hybrid simulation and human interaction with the evaluation software, as well as linking that software to other USAF C^2 systems, is discussed.

This study was supported by the Air Force Human Resources Laboratory (AFHRL), Wright-Patterson Air Force Base, Ohio, under Contract #F33615-87-C-0007 with BBN Systems and Technologies Corporation, Cambridge, Massachusetts.

We express our appreciation to the personnel at Det 1, 4444 Operations Squadron, Luke Air Force Base, Arizona, for their expert guidance pertaining to tactical air defense operations and tactical controller training. In particular, Major Larry Reed and Captain Curt Wright staffed special arrangements and worked extra hours in support of this effort.

We also appreciate the efforts of the United States Marine Corps at MACSONE, Camp Pendleton, California. Major T. J. Kirk made it possible for us to collect data on equipment modelled in the study. In addition, Lieutenant Swett, Lieutenant Ellrick, and Corporal Drebing assisted well beyond normal duty hours.

We thank Captain Robert Lander, Electronic Systems Division, Hanscom Air Force Base, Massachusetts, who provided the technical documents necessary for encoding equipment characteristics.

We also appreciate the Litton Corporation Air Force Plant Representative, Captain Jerry Fitzgerald, for supporting our data collection efforts on equipment modelled in the study.

Additionally, we thank the staff of the 102nd Tactical Control Squadron, Rhode Island Air National Guard, for sharing their technical expertise on Control Reporting Center operations.

The authors are appreciative of the staff at the 101st Tactical Control Squadron, Massachusetts Air National Guard, for allowing us to observe, collect data, and try our hand at "controlling" simulated tactical air missions.

We would also like to express our thanks to the staff of the USAF School of Aerospace Medicine at Brooks Air Force Base, Texas. We are particularly grateful to Dr. Sam Schiflett and his staff, Dr. Dave Strome, Mr. Neal Takamoto, Mr. Phil Tessier, and Mr. Mathieu Dalrymple for their help and cooperation in collecting verification and validation data on GCI controllers that are used in the validation phase of this program.

Finally, this work could not have been performed without the extensive assistance of the staff of the Air Force Human Resources Laboratory (AFHRL) at Wright-Patterson Air Force Base, Ohio. The Air Force program manager was Captain Eugene Henry. In addition, Major Donald Smoot provided operational expertise concerning Command and Control. Their cooperation and guidance, as well as their broad subject-matter expertise, have made significant contributions to the project.

TABLE OF CONTENTS

List of Figures

Methodology for Evaluation of Automation Impacts on Tactical Command and Control (C^2) Systems

Final Report

I. INTRODUCTION

This report describes the development and implementation of a prototype methodology to assess the impact of automation in the United States Air Force (USAF) Command and Control (C^2) systems. This C^2 analysis methodology provides a computer-based and model-referenced tool with which to examine automation's impact on operator performance. Object-oriented models of both equipment and personnel allow efficient and effective analyses to be carried out. These object-oriented models, in turn, have introduced a promising new research paradigm whereby an actual operator can interact with simulated operators and equipment.

Background on Problem

Impressive advances in the size and speed of computation environments have enabled the recent introduction of automation into areas of system operation traditionally the domain of human operators (i.e., decision-making, diagnosis, and control). Although technology has provided new and increased capabilities for operators in complex systems, the introduction of automation is generally accompanied by uncertainty. There is uncertainty about the impact of automation on a complex process such as that of Command and Control, the burden imposed on the operator to understand and work within the constraints of automated systems, the type of training required to make effective use of the "improved systems," the likely places at which the automation/human system will break down, and the nature of potential system errors.

The uncertainties described above are often unaccounted for in the design phase of developing human/machine systems. As a result, adding automation has frequently failed to fulfill the expectations of improved performance in human/machine systems. Major reasons for this failure are the lack of analytical techniques, human/system performance models, and evaluation tools that can be applied to full human/machine systems. Often, in practice, the functional capabilities made possible through automation have far exceeded the development of design and analytical methods needed to fully exploit these capabilities.

The general nature of the problem of assessing the impact of automation in human/machine systems is faced in a number of domains. Aircraft, both commercial and military, have seen the rapid and far-reaching inclusion of automated systems working hand in hand with the pilot or flight crew. Power systems, logistical support systems, communication and data networks, and process control systems have similarly been subject to changes in the way the operation of control is implemented. In all these instances, the function of the human in the system has undergone significant change.

We will explore some of the characteristics of automation development in complex systems to motivate the discussion of our approach to the methodology:

1. To be usefully predictive, an analysis methodology must be responsive to the general issues of human/system interaction wherein the technology of automation has invested the system with control and decision-making authority.

2. Intelligent automation is forcing humans into a different operational relationship with complex systems. Human operators are now more often called upon to be managers or partners in the systems which they operate.

3. As the point of system development moves from hardware- to software-based improvements, the pace at which development and testing must proceed is significantly increased. Standard evaluation and test of equipment effectiveness (e.g., equipment prototyping or system integration into full-scale exercises of operation) are costly, cumbersome, and too infrequent. The development schedules simply outpace the test and evaluation schedules. The analysis system must, therefore, be able to be adapted to new designs and new capabilities.

4. The improved connectivity of distributed systems for Command and Control suggests a new criticality in accuracy, as there are fewer check points in system operation. Striking testimony to this criticality in Command and Control can be found in the House Armed Services Committee hearing on the Iranian airbus 655 incident (H.A.S.C. No. 100-119, 1988). In this incident early misinformation propagated its effect through the C^2 decision process, leading to an international incident and personal tragedy. Communication and information flow must be explicitly represented in a C^2 analysis tool.

5. Finally, one issue associated with the initial introduction of automation into a complex system is that there is often no empirical experience with which to address design issues such as those suggested above. This lack of data can hinder the successful integration of the automated system because there is no basis of comparison between the new and current systems. Furthermore, this situation will persist until appropriate predictive and analytic tools are produced.

With these problems in mind, we are developing an analysis methodology to address the issue of automation impact in USAF systems. We have focused on C^2 operations and, specifically, on tactical defensive counter-air operations. Because the operational characteristics play an important role in system representation for analysis, we will now provide a brief description of the air control environments that have formed the testbed for our C^2 analysis methodology development.

Background on Tactical C^2

The USAF Tactical Air Control System (TACS) is responsible for the planning and execution of tactical (i.e., within an operational theater) air-to-air and air-to-ground operations. It consists of several echelons and elements. At the upper level is the Tactical Air Control Center (TACC), which is responsible for overall battle planning (issued in the form of Air Tasking Orders or ATOs) and for conduct of the deep strike missions such as air interdiction (AI) and offensive counter air (OCA). Subordinate to the TACC, and responsible to it for the conduct of the defensive counter air (DCA) or air defense mission, are several echelons of radar-based elements: the Control and Reporting Center (CRC), the Control and Reporting Post (CRP), and the Forward Air Control Post (FACP).

In developing a methodology for evaluating the impact of automation on tactical C^2 operators, the major characteristics of the tactical operations must be considered. We have selected CRC operation as our testbed because it captures tactical air control operations. For instance, the CRC is replete with decision-making, chain of command, communication exchange, and selection of courses of action that are the core of tactical C^2. These functions are highly affected by the goal states of the C^2 element and the individual decision-maker within the context of the tactical situation at the time of the decision. These tactical ground-controlled intercept (GCI) operations are highly procedural, but the selection of appropriate procedures is very situation/context-sensitive.

Some procedural decisions are based on semi-rigorous assessment of the situation; others are almost purely heuristic and based on recognition of a pattern or situation.

In addition, the CRC operations consist of typical activities supported by these procedures, such as manning Combat Air Patrols (CAPs), scrambling Friendly aircraft, and pairing. Manning the CAPs is simply a matter of ensuring that a set number of Friendly fighters are, or will be, orbiting a navigation point (i.e., CAP). Scrambling is the request to an airbase to have more aircraft become airborne to man CAPs or to engage hostile aircraft. Pairing is the assignment of Friendly fighter aircraft to a hostile aircraft for the purpose of visual inspection, escort, or attack.

C^2 Operational Constraints

The purpose of the C^2 analysis methodology described in this report is to serve as a tool with which USAF analysts can anticipate the effect of the introduction of automation into complex C^2 systems and to provide predictive measures of human performance in these systems. We feel that such data can be of service in the design of C^2 systems, particularly in specifying the training requirements for such systems and in guiding the acquisition of future systems.

Recent developments in the nature of C^2 operations and in the technological developments that support these operations provide additional constraints and requirements for the methodology developed to analyze their effectiveness. These recent developments in C^2 reflect several trends: (a) the need to make quicker decisions about an evolving tactical situation (a requirement that will become increasingly important as the number of assets available to respond to threats is decreased), (b) the need to process and integrate increasingly large streams of data from multiple sources, and (c) the need to make the currently large and relatively immobile C^2 facilities less vulnerable.

The introduction of automation into the C^2 GCI operation promises great increases in ground control capability, an improvement in mobility and modularity, and a decrease in the number of personnel required in a given area of responsibility. There are, however, a number of issues that attend the introduction of automated systems into this environment.

1. Will human workload saturate a particular system and will procedural bottlenecks be revealed? Given an ability to handle several times the number of radar

tracks that current ground control systems can handle, when do the human operators begin to reach performance limits and how will they offload excessive requirements?

2. What will the duty cycle or workload of an operator be in an automated system? What are the transient or peak loads that can be handled and what are the long-term strains that will be encountered? What are the procedural differences between current and advanced systems? On a broader scale, what are the tactical and doctrinal differences that the advanced control capability will impose on current TACS operational standards?

3. What is the impact of automation initiatives on manpower and training for new systems? Given the complexity of rapidly reconfigurable software- and firmware-based control systems, what are likely sources of operator error and what demands for special training will be incurred? A system could be rendered ineffective if operators are not able to exploit the full range of system features because of inadequate training.

4. What will be the effect of automation on the information and data requirements for system operation? In designing for optimum information flow, the designer must determine the paths and media through which to supply information. Automation provides the C^2 designer with flexibility, but raises new issues as to the form (semantics) and method (syntax) of providing operator data. The inevitable increase in data that automation provides must be balanced by design to avoid operator overload.

5. How can automation be effectively transferred into the TACS elements? How will personnel who are experienced with existing systems adapt to the new automated facilities and procedures? How will automated operations be integrated with non-automated systems? How will the system transition be implemented?

6. What are the procedures associated with system verification and validation? The introduction of automation raises new challenges for operational, integrated operator/system testing.

Methodology Description, Features, and Analytic Utility

It is our intention that this methodology help equipment designers, systems analysts, and operations mission developers to answer the above questions. Before detailing the rationale for the system's design and its implementation, we will describe the C^2 analysis methodology, its features, and its analytic utility, and the intended utility for designers and analysts of C^2 systems.

C^2 Analysis Methodology Description

The analysis methodology is a workstation environment that integrates the following feature such that a synergistic and more powerful analytic approach is realized for C^2 systems evaluation:

1. Development and manipulation of operational scenarios.

2. On-screen prototyping of candidate C^2 hardware.

3. Models of operator (and team) activities, performance, and responses.

4. Insight into model execution.

5. Data collection based on emulated operators using the on-screen prototype within the context of a given scenario.

We will describe each of these features in turn.

Scenario and Model Development

To exercise prototype equipment and force response from the human operators (or their model equivalents), the designer must have access to tools to generate a script. In the case of GCI C^2, such a script requires at least control over the placement, routes of travel, capabilities, and missions of Friendly and Enemy air assets.[1]

Figure 1 shows the types of scenario parameters manipulable through this interface tool.

Equipment Configuration and Capabilities

In addition to manipulating the force laydown which the GCI system must counter, the designer should be provided an ability to specify the capabilities of the equipment which the operators (or their model equivalents) will have to operate.

---

[1]As the complexity of missions increases, ground support and defense from missiles or guns would also be included, as well as airborne radar capabilities.

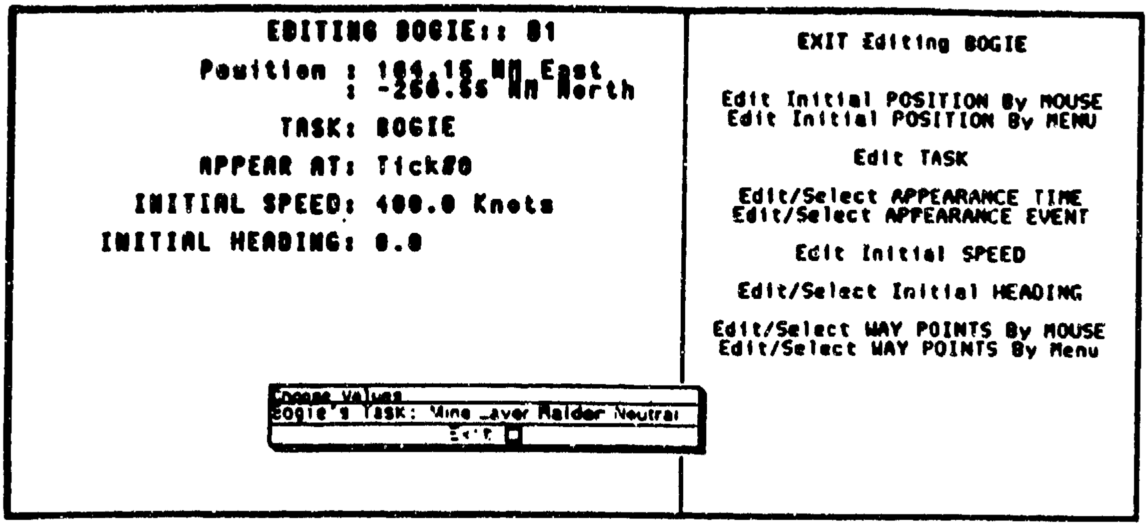


Figure 1. Aircraft Object Edit Menu. Editing Screens for Bogies.

Figure 2 shows a fairly complicated pattern of attack which is displayed by prototyped equipment and reacted to by models of CRC operators.

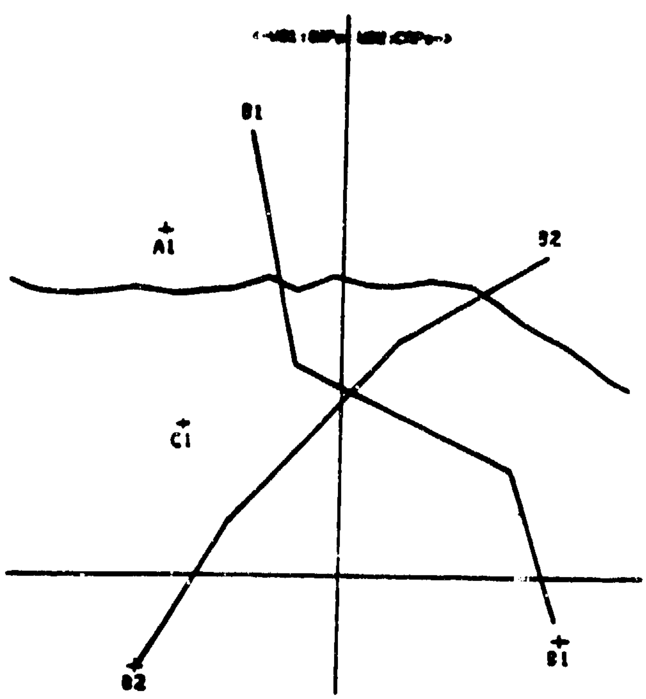


Figure 2. Example Bogie Tracks (B1, B2) Produced with the Scenario Generator. C1 & C2 Represent Combat Air Patrols; A1 and A2 Denote Air Bases.

Figure 3 shows the workstation display associated with the Modular Control Equipment (MCE) with which our operators and models will interact.

The particulars of the MCE equipment will be provided in later sections of this report. The point here is that a functional representation of the equipment is provided for the analyst. This visual "soft prototype" allows the analyst to see a representation of the operator's workstation. Further, when the system is "active," the analyst can effectively "look over the shoulder" of any of the operators in the MCE. The software design of this equipment emulation allows the analyst to move, resize, and redesignate the functionality of any of the MCE display and control elements. In addition, the assumed level of automation (i.e., the functions assumed to be performed with the black box of the MCE, such as pairing, intercept calculation, and identification of Friend, Foe, or Neutral) can be manipulated to account for more or less automation. The impact of automation and console procedures on performance can thereby be investigated.

Human Performance Models

Providing manipulable representations of the operational context and of the equipment that is to operate in that context is a first step in providing designers the tools to investigate automation impacts on C^2 operations. The more challenging task is to provide those designers with an examinable, consistent, and valid representation of the human operators who will be interacting with that equipment and responding to that scenario. To investigate the impact of automation on operators, we have developed representations of the human operators in the MCE. These models describe (within the limits of state-of-the-technology) the responses that can be expected of human operators in several areas critical to C^2 operation. Specifically, we model the human visual and auditory perceptual processes. In addition, we try to account for resource limitations in time, data, and cognitive capacity and we model human response in terms of both verbal and psychomotor output. These representations predict human performance based on the goals of the C^2 operation as constrained, or facilitated, by the particular equipment with which the operator interacts.

Understanding of Operator Behavior

The system provides designers with explicit and examinable references to the rules, decision-making strategies, heuristics, and assumptions under which the full C^2 human/machine system is operating. This gives the designer a unique ability to examine,

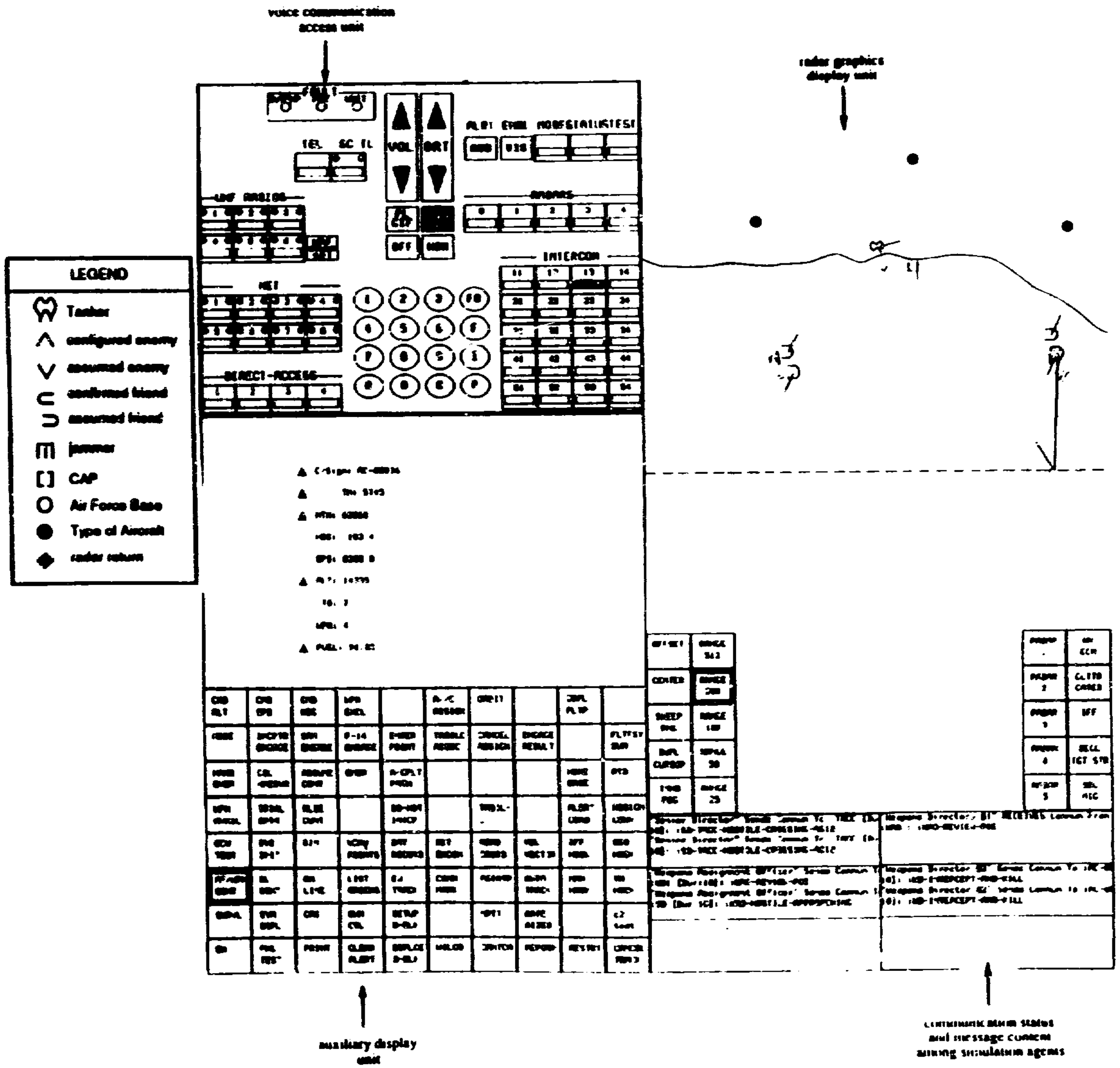


Figure 3. Layout of the Workstation Display Simulation of the MCE Equipment.

at any point in the simulation, the cognitive state of all the MCE operators, the rules being used to guide their behavior, and their nominal workload. Direct manipulation of the cognitive state allows the designer/analyst to obtain answers to "what if" questions about how critical a rule or a piece of information might be in a given mission context.

Performance Data/Analytic Capability

As the simulated operators interact with the equipment emulation in response to mission demands, data can be collected about that performance. Each action taken, decision made, and communication by the MCE crewmembers is logged by the analysis system. Subjective estimates of the task load are associated with each activity, and are also logged as data. These data are logged for each operator in the simulation each 500 msec. An analyst, then, has available a full record of operator-model performance that can be examined and manipulated to meet his/her experimental requirements.

In summary, the methodology provides the designer/analyst with a tool to manipulate the operational environment, the equipment characteristics, the assumed human performance requirements, and the data collected for various test runs. The methodology is designed to be robust in the face of changes in the above characteristics. It is also designed to be used, modified, and manipulated without recourse to the level of code/program interaction. The user-interface has been designed to facilitate limited system exercise by domain experts rather than computer scientists. The analysis system, like the user interface, is in a prototype stage of development, therefore, the ideal usability has yet to be realized.

II. OBJECT-ORIENTED MODELLING ENVIRONMENT

It has traditionally been difficult to predict the impact of prototype and developing systems prior to fielding and testing. The current effort attempts to address that difficulty with a predictive evaluation simulation methodology to determine the impact of introducing automation into a complex network of responsibility such as TACS.

The general nature of the problem, to develop an analytic methodology for a rapidly changing and semi-autonomous human/machine system, suggests the following considerations:

1. The analytic methodology must be modular to incorporate and respond to evolution in particular subsystems without jeopardizing the integrity of the full analytic structure.

2. The analytic methodology must be extensible to respond to the development of new capabilities or facilities within a system.

3. The system must be able to describe and predict the human operator's performance consistent with the level of autonomy exhibited by the system.

To respond to the requirements which both the domain of interest and an aggressive automation development cycle impose on performance evaluation, we have developed an architecture for our analysis methodology that is modular, extensible, reconfigurable, and able to represent the behavior of intelligent agents, both human and computational. These principles have guided the overall system structure, and they were also instrumental in our selection of the software development environment used in this project.

We have selected an object-oriented programming paradigm to implement our analysis methodology which includes the description of the human operators, the MCE equipment, and the scenario of operation.[2] This approach is contrasted with traditional programming techniques in which programming consists of directing the flow of logic through a series of procedures. In object-oriented programming, one programs by first describing types of objects in the simulation world of discourse and then describing their internal state and the procedures they are to carry out.

The objects in the simulated world are of varying types, which inturn belong to various classes. For example, in the MCE, some objects are "agents." Of these, certain of the agents are of the type "human-agent." These human-agent objects share the operational characteristics that are common to the class "human-operators."

The objects in the world have state information that is stored locally with the agent, (e.g., in the case of human operators, part of that state is what each operator knows about the condition of the airspace that defines the scenario for GCI). Objects communicate with other objects in the simulation through a convention of message-passing. Objects have procedures that specify how they are to communicate and with whom. They know how to compose and receive messages. The objects in the world also have procedures that are to be carried out when a particular message is received from another object in the simulation.

---

[2]Specifically, we have implemented the system in ZETA, LISP, SYMBOLICS Genera 7.2. As the Common LISP Object System (CLOS) has become available, we have upgraded the software to be compatible with this development.

Developing an object-oriented simulation consists of describing the objects that are relevant to the simulation environment, creating instances of them and causing them to interact. For certain classes of problems in representation, the object-oriented methodology is much more suitable than traditional, procedurally-defined programming methods. The earliest application of object-oriented programming was to carry out simulations (Birtwhistle, Dahl, Myhraug & Nygaard, 1973). Simulation programming continues to be a domain for which object-oriented programming is well suited, as high-level, simulation is a representation of a group of objects and their interactions (Goldberg & Robson, 1983; Steele, 1988). Further, as will be discussed below, the object-oriented simulation methodology uniquely addresses the considerations specified above.

In response to the need for modular development, object-oriented methods enforce modularity through the definition of the boundaries inherent in object specification. This modularity is maintained throughout the higher-level constructs of the simulation environment.

The evaluation architecture is open in that the descriptions of simulation entities and performance models are not considered to be complete or exhaustive of the simulation system's capabilities. It is extensible in that more exacting model formulations (for example, of human performance) can be integrated into the structure without perturbation of the other descriptive models of the environment, the scenario, and the equipment. Further, the methodology is compatible with an increase in the scope of the operations performed (e.g., battle management or TACS functions are clearly available as an extension to the current representational scheme).

The object-oriented structure is relatively easy to modify or amend, given its modular characteristics. Changes in the assumed human/machine functions, for instance, can be supported without reconfiguration of the entire system's architecture. For instance, the level of automation in aircraft identification Friend or Foe can be varied from a fully automated task to a task that rests completely on the surveillance supervisor. Further, the object-oriented methodology supports modification through the process of inheritance whereby members of a given class inherit features, or characteristics, of more general descriptions of that class. For instance, if the process by which communication is effected among operators is changed, then every instance of that communication process will be changed automatically to reflect the new protocol.

The simulation technique supports description of the human and machine function in similar terms (i.e., that of intelligent agents). The particular characteristics of human or machine intelligence are simply specialized instances of the more general class of intelligent agents. Assignment of a particular task to either a human or an automated agent (e.g., an identification or decision task) can be made at simulation run time, as the basic task is described in terms applicable to intelligent agents.

Finally, the object-oriented approach provides structural and organizational information that a more traditional task-analytic representation of the domain lacks. The process of organizing information by types and classes, as well as assignment of relative standing to objects in terms of generality or belonging (i.e., the specification of the inheritance protocol), in effect determines the relations among concepts in the simulation domain.

The evaluation system is organized in terms of components that have face validity in the real world. For example, the MCE is composed of four systems, the radar graphics display unit (RDGU), the auxiliary control panel (AUX PANEL), the voice communications access unit (VCAU), and the control panel assembly. These are represented as objects in the system, and the component structures of these systems are similarly represented as component objects in the software. Similarly, actions taken are arranged in a hierarchy whereby high-level goals are composed of subgoals and tasks.

In standard functional or timeline analyses the organizing principle is exclusively time or precedence relations. Although these relationships are included in the analysis, system, the domain of operation is also organized in such a way that it can examined and manipulated by the designer/analyst.

In addition to these general requirements, several characteristics of the Command and Control process must also be accommodated, for the ultimate success of this methodology depends on its appropriate representation of the salient characteristics of C^2.

Specific C^2 Requirements for the Modelling Environment

Tactical Command and Control is a cyclical decision and communication process. It is performed under tight time constraints, and may be considered to have four steps: (a) situation assessment, (b) development of a course of action, (c) execution of that course of action, and (d) feedback of the results of that execution. These steps are familiar to the human performance analyst and have been applied to the analysis of other dynamic and interactive systems (Corker & Baron, 1989). However, the C^2 operational environment, in which the operator is interactive with adversarial forces, introduces significant complexity in predicting and evaluating operator actions. The C^2 tactical ground controller is faced with an open and unpredictable environment when trying to counter an intelligent enemy who is intent on negating or destroying the control center and the forces under its control.

The complexities of the C^2 environment are also inherent in the distributed nature of its operation. Teams of GCI C^2 operators must hold a common perception of the existing situation if they are to achieve a coordinated response. This requirement for combined situation assessment and interaction compels a concern for automation that supports distributed decision-making among C^2 operators as a team, as opposed to a system that allows only a sequence of independent operations. The implication for C^2 operation is that although each crewmember individually performs his/her own assessment of force capability, situation, and most likely Friendly or Enemy intent, the crewmembers need to communicate with each other to provide commonality to their assessments. The course of action each operator chooses should be the most appropriate for that operator's area of responsibility, based on the mission goals and the constraints of coordinated action. The modelling requirements are made more difficult by the requirement to maintain and track that communication exchange and world view development.

Specific C^2 Characteristics

The specific characteristics required for a realistic model of the C^2 environment include decision-making, taking account of chain of command, course of action selection, and communication exchange. These form the core of tactical Command and Control, and are highly affected by the goal state(s) of the particular C^2 element and individual decision-maker within the context of the tactical situation. Each characteristic is described in greater detail below.

Decision-Making. Many tactical C^2 operations are highly procedural, but the selection of a certain procedure is very sensitive to context and situation. The selection processes and procedures an operator might use to arrive at a decision are not always the same and may be based on the following methods: a semi-rigorous assessment of the situation, a purely heuristic approach, and/or pattern recognition.

Chain of Command. Tactical C^2 has a strong hierarchical aspect. A hierarchy exists among C^2 elements, among operators within a given element, and even among the goals and activities of a given operator. The manner in which these hierarchies function together is rarely straightforward. For example, personnel usually carry out their supervisors' directives; however, the hierarchical influence is preempted when the decision is derived from a subordinate's more perfect knowledge and understanding of the situation.

The command structure of the Control and Reporting Center (CRC) equipped through MCE is as follows:

Battle Management Personnel. The Senior Director (SD) is assigned as the top-level battle management decision-maker. The SD implements the TACC's directives and delegates authority, applies/interprets the theater Rules of Engagement (ROE), and employs assigned air defense resources.

Surveillance. In the current 407L CRC system, the aircraft detection and tracking function is supervised by an Air Surveillance Officer (ASO), and the process of determining which radar returns are aircraft and which are noise is performed by human operators known as Search Scope Operators (SSOs). When the SSOs decide a return is an aircraft, they must perform a number of steps to initiate a computer track of the aircraft. In the target system, an MCE-equipped CRC, this function is supervised by a Surveillance Supervisor (SS). Because this detection and tracking function is almost completely automated in the MCE system, there will be few, if any, SSO personnel.

Identification. The ASO or SS also supervises the identification (ID) function, sometimes also called "movements and identification." This function uses a number of electronic and procedural methods to make Friend-Foe decisions on the "pending" tracks generated by the surveillance function. Aircraft identified as Friendly generally require no further actions. In some cases, there are insufficient data to make a definite decision and the track is classified as an Unknown. Unknowns have subtypes such as Assumed Friend and Assumed Enemy. The procedures for dealing with Unknowns differ between peacetime and wartime environments and are affected greatly by the theater's ROE. The ROE strongly control the decision to declare an aircraft a Foe or Hostile. As noted earlier, ID technicians in a 407L CRC expend much energy performing the basic electronic and procedural checks. In an MCE-equipped CRC, these checks are performed very quickly by the computer-supported ID algorithms. Thus, instead of starting with the tracks in pending status as the 407L ID operator does, the ID technician in an MCE CRC generally finishes identifying tracks the system has previously classified as Unknown.

Weapons. The weapons allocation function in a CRC is performed by the Weapons Assignment Officer (WAO), who also supervises the activities of two or more Weapons Director (WDs). The WAO and SD develop and implement a fighter employment strategy that includes a plan for balancing the number of aircraft on airborne alert (usually on combat air patrol or CAP) with those in various alert states on the ground. The WAO must also decide, based on the rules, when to "scramble" (order for immediate takeoff) the ground alert aircraft and when to return-to-base (RTB) airborne aircraft. Typically, the WAO directs the location and assignment of aircraft to the CAPs, the use of any airborne tankers, the assignment of areas of responsibility to each of the WDs, and the use of available radio frequencies. The WDs provide voice directives/information to the fighter pilots and execute the WAO's fighter employment strategies. One of the WDs may also be controlling an airborne tanker aircraft to provide aerial refueling of the fighter aircraft.

Courses of Action. Conflicts among or within the hierarchies, as mentioned above, tend to promote frequent interruptions of activities. In some cases, the interrupted activity is resumed at the conclusion of the interruption. In other cases, the interrupted activity is restarted. Indeed, in some instances, the interrupted activity is entirely forgotten. Personnel normally try to achieve goals they have been given, but the situation dictates what course of action has priority.

Communication Exchange. A significant portion of communication activity focuses on the exchange of information about the context and situation to ensure that distributed individual decision-making is optimized. This information-sharing to attain a common situation assessment is of critical importance to the operators and has two main characteristics. First, substantial amounts of information are exchanged, often without producing any observable result. Initially, operators may not recognize a pattern requiring action. Moreover, they many not actually start an action. However, the operators are adding to or confirming their internal representations of the tactical world. Second, this information exchange and the creation of a common understanding of the situation are likely to be sensitive to the type of facility, communication channels, and operational environment. In particular, the modularity and physical dispersal characteristics of new C^2 system designs such as MCE are certain to have a significant impact on communication exchange.

Modelling Environments for Representing Human Performance

In addition to capturing the characteristics of C^2 operation, the simulation requires efficient and effective representation of the significant portions of human behavior and their interactions with system performance parameters. The user/analyst must have the capability to examine and manipulate the analytic system's models of its components. These model entities must include the mission being performed, a description of the operators performing it, and a description of the equipment being used in its performance.

Until recently, human/machine simulation has required extensive and exacting representation of each of the components of the system under study. The work and costs associated with full-mission simulation development, encoding, and execution often severely constrained the type of investigation that could be supported. These costs and the difficulties associated with scenario development also limited the breadth of factors investigated in simulation studies. Further, simulation techniques for human operator performance have tended to concentrate on the "micro-behavior" level of elementary task performance. This concentration was driven in part by the lack of empirical support for the validation of more complex behavioral analyses and in part by the constraints in current modelling languages. Human model outputs had to be expressed in terms that were mathematically tractable to the system description. In choosing the development path for the present methodology, we considered other modelling techniques, several specific simulation languages, and stochastic-based operations research methodologies as potential candidates to support the analytic workstation development. These are described below.

Simulation Languages

Recently, several simulation methodologies have been developed (Chubb, Laughery, Pritsker, 1987); however, these general-purpose simulation languages and special-purpose modelling frameworks impose certain constraints. For example, a block-diagram modelling system such as the General Purpose Simulation System (GPSS) provides fairly easy construction of simulations by linking functions which are contained within primitive boxes (Schriber, 1974). Primitive boxes encode low-level human performance functions such as short-term memory loss or human reaction time in a particular situation. However, GPSS is limited in the range of functions and the resolution of processing. GPSS also runs slowly because of the low-level definition of its primitives. Such characteristics are not suitable for the functionally complex nature of tactical C^2.

The technique of network transition models requires pre-definition of all possible paths through the network in order to operate. Simulation Language for Alternative Modelling II (SLAM II) is an example of a modelling technique that provides a network structure and symbol functions with a pictorial programming capability (Pritsker, 1984). Though discrete and continuous functions are provided for in the main code, complex event routines, such as transition and selection logic, must be encoded separately. To provide this logic, the user must exit the principal SLAM II network and encode the logic in another language, FORTRAN. Because tactical C^2 simulation would require encoding of complex logic, the need for frequently exiting the main program would offset the benefits of using this simulation language. In addition, the network (i.e., sequential) trait of SLAM II does not readily lend itself to the variable paths of activities and the loosely defined communication exchanges common to tactical C^2.

Although SLAM II does not offer intrinsic provisions for human models, the Human Operator Simulator (HOS) approach does pre-define elemental human operations such as movement and information intake. HOS also provides a language in which to write procedures that call these elemental processing functions into play (Lane, Strieb, Glenn & Sherry, 1981). However, as a pre-packaged tool, HOS imposes constraints on the number of operators that can be modeled, as well as having only a limited number of operator activities defined. These conditions are too restrictive for a tactical C^2 environment with virtually unlimited activities being performed by many operators.

Stochastic-Based Operations Research Techniques

A common operations research technique for simulating procedural environments is that of stochastic processes. This technique, which commonly resembles a network, involves drawing a probability from a distribution to determine the outcome of a particular event. Three difficulties surface, however, when attempting to adapt this framework to tactical C^2.

First, in some procedural environments like a production line, it is possible to collect statistical data and apply the technique of using probability sampling within a sequence. Indeed, tactical C^2 does involve procedural sequences. However, unlike highly proceduralized environments such as a production line, procedural sequences in C^2 depend on the situation and can be interrupted, compressed, or deleted if the operator approaches overload and/or has higher-priority actions to perform. Furthermore, because the distribution-based predictions that dictate the guidance of the sequence are contextually dependent, it would not be possible to specify, in advance, the relative likelihoods of the procedure selection at a particular branching point. Moreover, distributions are based on a particular, static situation, but the tactical environment routinely changes in a dramatic fashion. This dynamic nature would require changing the probabilities for all the various sources of action.

Second, the method above assumes that the analyst has collected and developed the probability distributions. Unfortunately, studies of distributions based on C^2 decision-making are rare or nonexistent because: (a) tactical C^2 is highly complex, and (b) few analytical tools are available to collect the data.

Third, the difficulties described above are further compounded by outcomes based on behavioral probability distributions intended for the occurrence of a given, unique event. The difficulty results from the large amount of decision-making (IF-THEN inferencing). Because a C^2 event is actually composed of multi-layered conditional probabilities, it cannot realistically or practically be considered unique.

Although well established, these stochastic-based human representation schemes are not well suited for representing a team of goal-oriented C^2 decision-makers. Key components of these techniques are *a priori* knowledge and predefined event- or task-oriented paths. Analysis of tactical C^2 reveals that these elements are not easily identifiable; hence, a requirement exists for new techniques and human representation schemes.

III. SYSTEM ARCHITECTURE

In describing our evaluation methodology, a discussion of system architecture is essential to understanding how the system is applied to support the evaluation of the impact of automation on C^2 systems. First, we will describe the modules of software that support system development. Then we will describe the control flow among modules providing the system connectivity. Finally, as an illustration of the use of the system, we will describe the data flow through the system that supports performance data analysis.

Modes of Operation

The C^2 evaluation methodology has been implemented to operate in two modes. The first mode of operation provides analytic and predictive performance data based on human and system simulation models. We have termed this the "analytic mode," and it is the basic mode of operation for the Automation Impacts Research Testbed (AIRT). The second mode supports operation of the C^2 evaluation workstation in a hybrid mode of simulation; (i.e., with concurrent operation by software representation of the MCE crew and a human operator). We have termed this the "manned-simulation mode," or Human-In-Process (HIP).

The C^2 evaluation system has been implemented in a modular fashion such that, in addition to these standard operational modes, external inputs and subsystems can be easily accommodated to support integration with other external independent systems that are part of USAF C^2 operations.

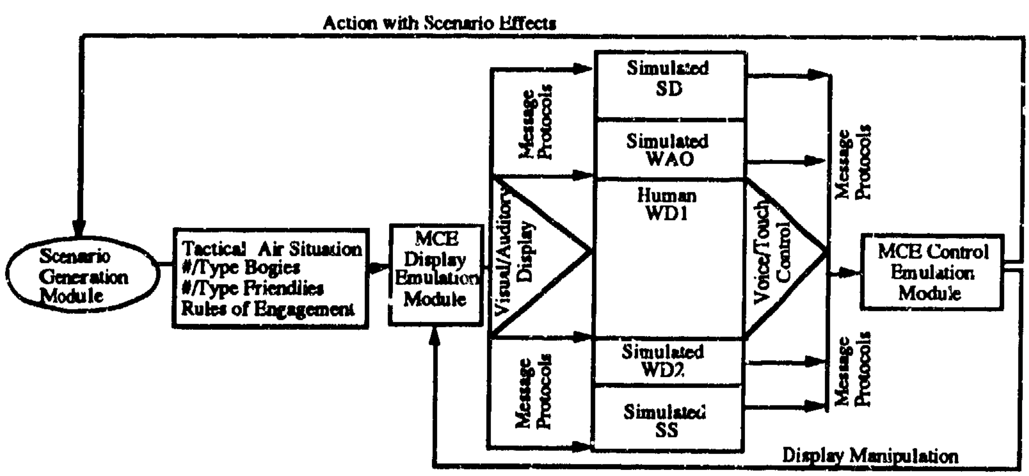
AIRT Mode Operation

In the analytic mode, the system is driven by a simulation scenario. A scenario is composed of events and objects to which the MCE crew-objects will have to respond. It can be considered like a script orchestrating the Enemy events and forces and tactics to which the Friendly forces must react. The designer is given several tools with which to design this scenario using screen-based menus.. The designer can select among Enemy aircraft types and give them individual flight paths into Friendly territory. He/she can select the number and position of Friendly air bases and CAP points. The number and type of Friendly aircraft can also be specified. Events such as Enemy attack can be scheduled to occur at specific times or be inserted into the running scenario at the designer's discretion. Although the scenario can be specified beforehand, the response of the Friendly forces is still made on a contingent and discretionary basis and is driven by human performance models.

The responses of the operators of the MCE to the analyst-specified scenario are generated by models of human performance that are tailored to individual operator responsibilities. These models respond to the stimuli presented to them via emulation of the MCE operation control module (OCM). The human operator models process incoming information, interact with the simulated MCE equipment, and communicate with each other via simulated message traffic. Performance analysis is provided through operation and interaction of the human operator models with the OCM emulation. These models provide prediction of individual operator response to the simulation scenario in terms of actions, perceptual response times, and performance accuracy. Execution of the selected action is modeled through motor response times and accuracies. Additionally, aggregate workload estimates are provided in terms of visual, auditory, cognitive, and perceptual-motor resource utilization. These predictions reflect data from the literature where available and otherwise reflect careful analysis and subjective estimation of elemental component performance specifications. The effect of operator action is "displayed" through appropriate response of the MCE equipment simulation. The analytic mode of operation also generates functional interactions among operators and mimics the procedural requirements for communication and data exchange. Each operator's rules of behavior are modeled according to his/her function in the MCE environment. This modelling includes duty assignments and protocols for interaction among the operators.

<u>HIP Mode Operation</u>

In this mode of operation, the C^2 workstation, through its emulation of the MCE equipment, can be used to provide input from an actual human operator to the simulation scenario and to support the interaction of that operator with the simulated operator objects. The design for this integration includes a voice recognition system to interpret the human operator's commands, a speech generation system to provide auditory output from other MCE operator objects, and dual screens with touch panel overlays to emulate the operation of the MCE control and radar graphics units. Figure 4 illustrates the control flow supported by the current implementation.



<u>Figure 4.</u> **Human Operator Interaction with the MCE Simulation in a Manned-Simulation Mode.**

The current functionality provides for switch actions to be taken by a human operating as a WD. The configuration of the system hardware is provided in Figure 5.

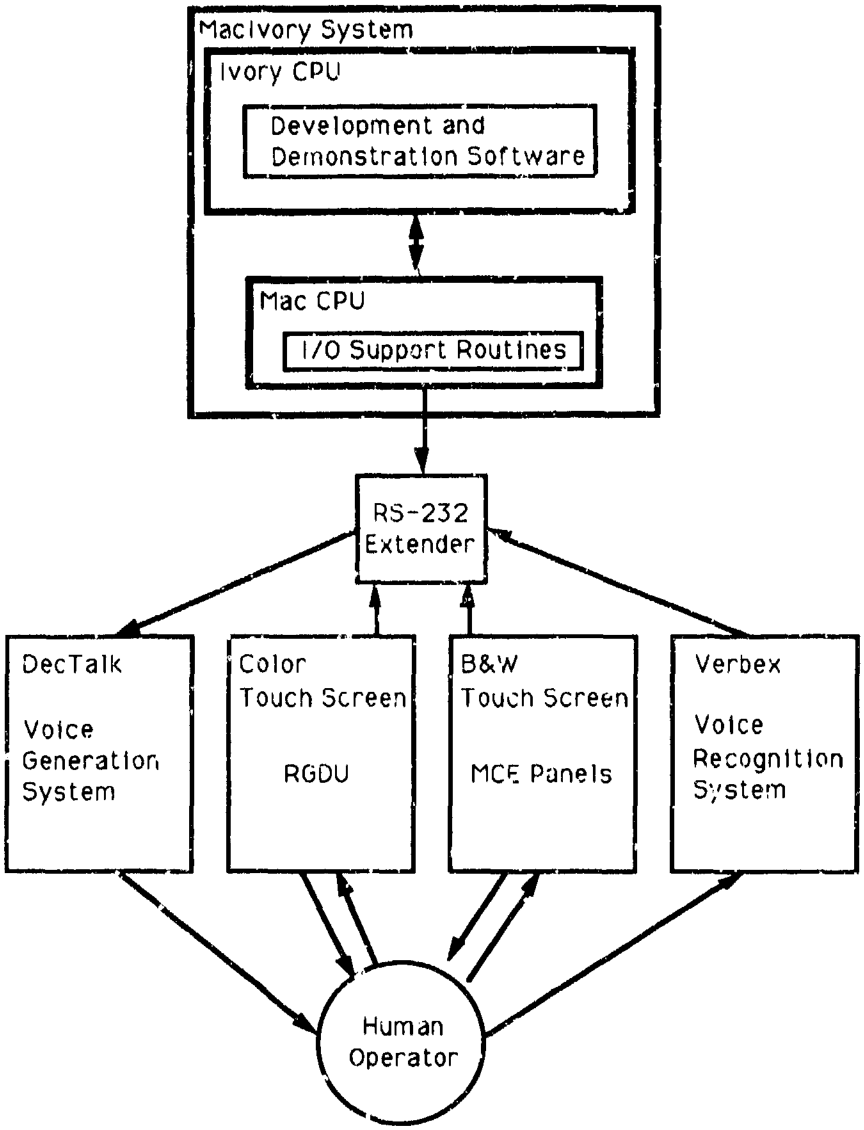


Figure 5. Hardware Configuration of the HIP System.

The Human-In-Process (HIP) operation of the evaluation system highlights some of the advantages we feel can be realized with object-oriented, model-based human performance modelling. The modular nature of the operator representations allows us to "remove" one of the operator models and to replace that function with an actual human operator. The formalization of the interface (referred to as a message-passing interface) among the objects of the system allows us to define a protocol for the interface between the behavior and vocalizations of the human operator and the underlying model-based system. As in actual MCE operation, we have provided a touch screen for user input into the MCE control system and for designation of aircraft and objects on the RGDU radar screen. In this way, an operator can access MCE functions, designate targets, select communication channels and modes, and make pairings. The actions taken by the human operator in the system are then communicated to the underlying model-based representations of the other operators in the system and to the aircraft that are directed to intercept.

We have provided further integration of the human operator into the system by including a voice channel input. For this input, we have developed a standard syntax and vocabulary with which the human operator can direct intercepts using actual voice control over the modelled aircraft under his/her control. To complete the integration of the operator with the system, we use a VERBEX voice recognition system to support oral communication between the modelled operator and the human operator in the system. (See Figure 5.)

We foresee several promising applications for the Human-In-Process mode of system operation. An immediate application of the HIP system is verification and validation for the operation of the modelled operators. If the human operator can react to the same scenario with behaviors similar to those of the modelled operators and can interact with those modelled operators to act as a team member, then there is some assurance that the analysis system is performing within a reasonable range of validity.

A second application of this mode of operation has, we feel, the potential for significant impact on Air Force operations. The system can be used as a training system for individual operators. Rather than having to assemble a full MCE crew to train a single operator, the human trainee can train against the modelled operators' behavior in an environment that allows rapid and easy reconfiguration of the team and of the scenarios being used.

Software Modules

The C^2 analysis methodology is composed of a number of independent but cooperating software modules. The considerations which led us to implement this distributed system development are the same as those which guided our selection of an object-oriented software paradigm; namely, modularity, expandability and modifiability. These software modules are illustrated in Figure 6. They consist of the following broad classes:

1. A basic set of system support tools and utilities. These are collectively called the LUDD system and include window system support, system creation and maintenance tools, core activities, and inference engine support.

2. A broad set of facilities. These go by the name BRAHMS and support the behavior representation and human modelling systems. Generally speaking, these are software modules built as enhancements, extensions, and specific instantiations of certain of the LUDD features. The following facilities are among those in the BRAHMS system: specific window-system applications that support displays of the human models' responses and behaviors; enhancements of the activities system to support the human modelling behavior; and the basic structure of the software components common to the BRAHMS system.

3. A top-level set of components. These are components specific to either AIRT or HIP, as well as those features that support C^2 projects in general. For example, included here are activity rules that are specific to the behavior of the human models in the AIRT/HIP systems and simulation support systems that are specific to the emulation of the MCE.

We will describe each of these systems in detail.

LUDD System

The LUDD system is a set of packages of underlying utilities and common systems upon which the higher-level features of the BRAHMS-based applications systems are built. This section describes the two major roles/components of the LUDD system -- the MASTER component and the SYSTEM LOAD component -- and some important and commonly used minor features.

The MASTER Component. In specific applications systems (such as AIRT and HIP) built on top of the LUDD system, there is typically a central software module known as the MASTER. This MASTER functions primarily as a channel of communications among the various modules that make up the complete system, and between the system and the "outside world" (a user or other software system running concurrently with the system). For example, in the HIP system the HIP-MASTER is responsible for passing communications among the various internal component subsystems. The HIP-MASTER passes communications among the human operator interface subsystem, the other human performance models that generate operator behavior, the BRAHMS-like displays, the HIP system, and the operator running the system.

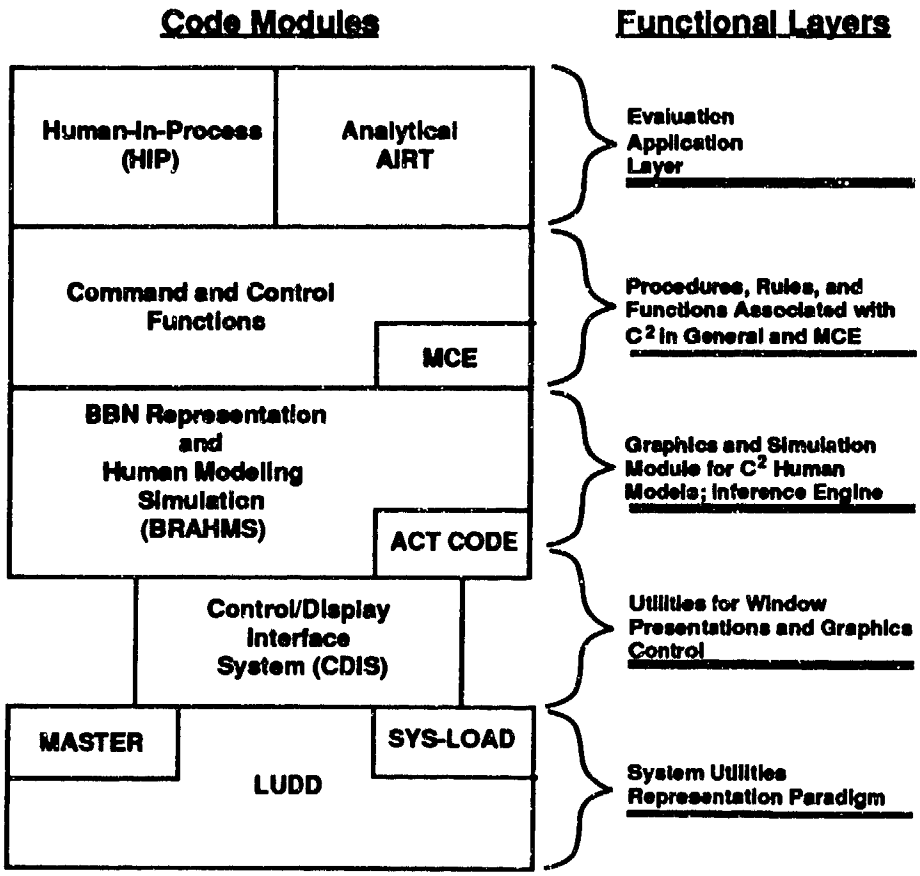


Figure 6. **C^2 Evaluation Methodology Code Architecture.** Following our basic modular philosophy, the C^2 evaluation system is supported by several functionally distinct modules. The top layer represents the two current applications for the evaluation, the human-in-the-loop operation and the basic analytic tool operation. These methodological applications can be expanded based on the support of the underlying modules.

One feature of being a "highest-level" component is that the MASTER is allowed to play a primary function in the creation of the system. The modules of a LUDD-based system, such as the HIP system, are typically laid out in a natural tree-like structure. (This is particularly relevant at creation time but may not be required at run time.)

SYSTEM LOAD Component. This tree-like structure, rooted in the MASTER object, allows a very straightforward mechanism for the creation and setting up of the entire system. In short, each node in the system tree is responsible for the (recursive) creation and initialization of its subsystems. Creation and initialization are accomplished in several stages, or passes.

An initial set of passes is responsible for making sure that all the modules of the system are created and in place, and that objects have the appropriate pointers to any other objects that they need to know about. Once the system is set up, with all objects and components in place, the system's initial state can be set. The first set of passes ("wire-up") or creation passes normally, need to be performed only once. The initialize pass can then be performed as often as is necessary to "reset" the system.

The CDIS Component

The Control Display Interface System (CDIS) is responsible for the displays used in a LUDD-based system (and, consequently, the windows on which they appear). The two most important types of objects in CDIS-based displays are PWINDOWs and Display CONTROLLERs.

PWINDOWs (Pseudo-WINDOWs) are basically "wrappers" which encase the actual machine-specific windows native to the specific hardware/software platform on which the application system is installed. All machine-specific features of the system are internal to these objects; consequently, the nature of the hardware platform should be transparent to any higher-level parts of the system. That is, the rest of the display system draws to these PWINDOW objects in a way that is independent of the hardware platform; therefore, changing the underlying platform involves changing only the kind of PWINDOW that the system uses; such a change will be transparent to the rest of the system.

Associated with each display in the system (each display usually encompasses an entire window) is an object known as a Display CONTROLLER. As its name suggests, this CONTROLLER is responsible for maintaining the state of the display and for "knowing" how to display the various features.

A given subsystem in the C^2 evaluation methodology system can have several displays that it must show. Here the term "display" is used to mean any self-contained image (a table, graph, button-grid, radar-screen, text output, etc.) that the system can output to the screen. A display typically has a dedicated output window, although this is not required.

The Display CONTROLLER governs all aspects of the display, including simply drawing the display, updating the output as the values of the program using it change, and refreshing the present state of the display in response to specific refresh commands or as the display appears or reappears on the screen.

Furthermore, any information about mouse-clicks or mouse-motions that occur on the window is transferred to the Display CONTROLLER currently governing that window. If no Display CONTROLLER is assigned to that window, inputs such as mouse-clicks, etc. are ignored. Thus, the Display CONTROLLER is free to respond to such clicks or motions in a way that is appropriate for its display.

Similarly, all keyboard input is passed along to the system itself, which is responsible for routing the input data to the CONTROLLER for the display that it currently has selected to receive the input. The CONTROLLER then uses the data as appropriate; for example, as commands to the system or as system input (e.g., as data entry in a table).

A Display CONTROLLER processes its graphics commands by acting on an object called its "PWINDOW" (i.e., a Pseudo-WINDOW). This object contains all the information specific to the windowing system on the underlying hardware platform system in use. The PWINDOW is responsible for translating any of the standard set of graphics messages that it can receive from its parent Display CONTROLLER into a command format appropriate for the hardware-platform-specific windows on which the application is currently running.

In keeping with our design goal of system modularity, the Display CONTROLLER paradigm provides that the physical platform-specific window is maintained by a simple, passive display device. The PWINDOW has no application-specific role. It does not reference any of the operational details of the application whose display it contains. This application-independent operation has two exceptions. These are the transferal of mouse-click information and the notification of keyboard entries. In these cases, the PWINDOW must pass information about what type of mouse-click has occurred, and where or what keyboard entries have been.

More precisely, all that is actually required of a platform-specific window is

1. that it can handle a standard set of output commands (e.g., a graphics command like DRAW-LINE/CIRCLE/RECTANGLE) and string-output commands, and

2. that it is capable of "remembering" and keeping track of which Display CONTROLLER (if any) is currently executing output on it, so that

3. it can transmit to its corresponding Display CONTROLLER data about the mouse-clicks (and possibly moves) that it receives, and

4. it can pass along keyboard information by, for example, placing any characters it receives into a common input queue.

Example of Use of the Display CONTROLLER Paradigm. As a specific example, we will consider the C^2 evaluation methodology's models of the workload and performance of a crew operating the MCE system. There are two major clusters of displays in the system; these will be described in detail in the *System Operational Concept Document* (Corker, 1990a) and *Software User's Manual* (Corker, 1990b).

One set of displays is a full-color representation of the console of the MCE system, as shown in Figure 7.

This display has two parts: a radar screen and a complicated set of touch-screen-activated pushbuttons and button-related panels. On the radar screen are a number of display icons representing the controlled aircraft and the symbology assigned to the aircraft by the MCE system. Associated with each aircraft is a Display CONTROLLER which is responsible for showing the aircraft's symbology, its radar return, various textual displays associated with the aircraft, highlighting/markings that the MCE system can associate with the aircraft, etc. A Display CONTROLLER is also associated with each grid of buttons on the MCE console display. Each Display CONTROLLER is responsible for showing the buttons in the Display CONTROLLER's corresponding grid, displaying the button in each of its various possible pushed states, accepting mouse-clicks on the button, etc.

The second is a monochrome display set showing the current workload and state of the human crewmember models, as depicted in Figure 8. Each crewmember model has two displays associated with it:

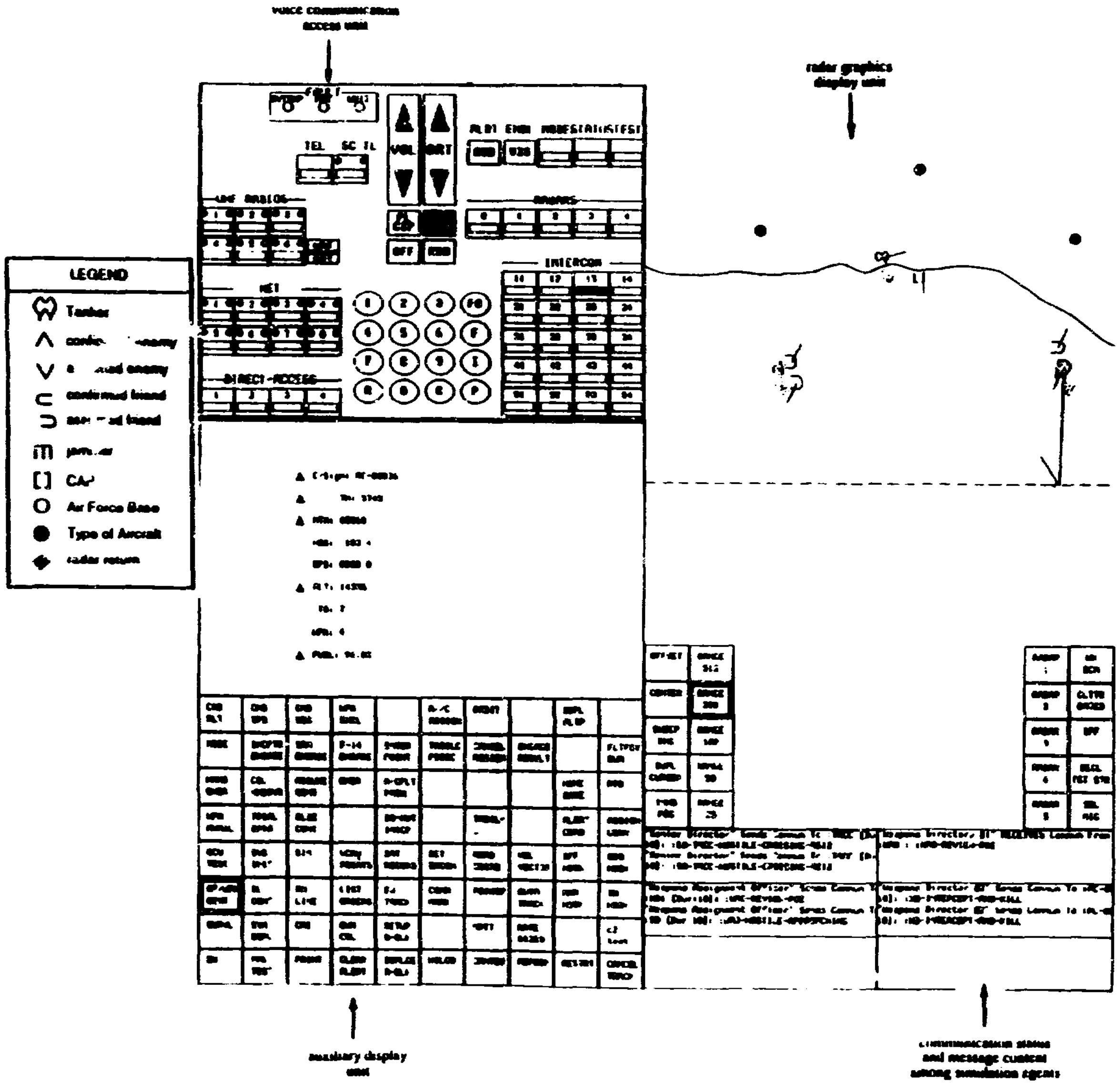


Figure 7. Color C^2 System Display. Explanation of the Color Screen Layout of the C^2 System.

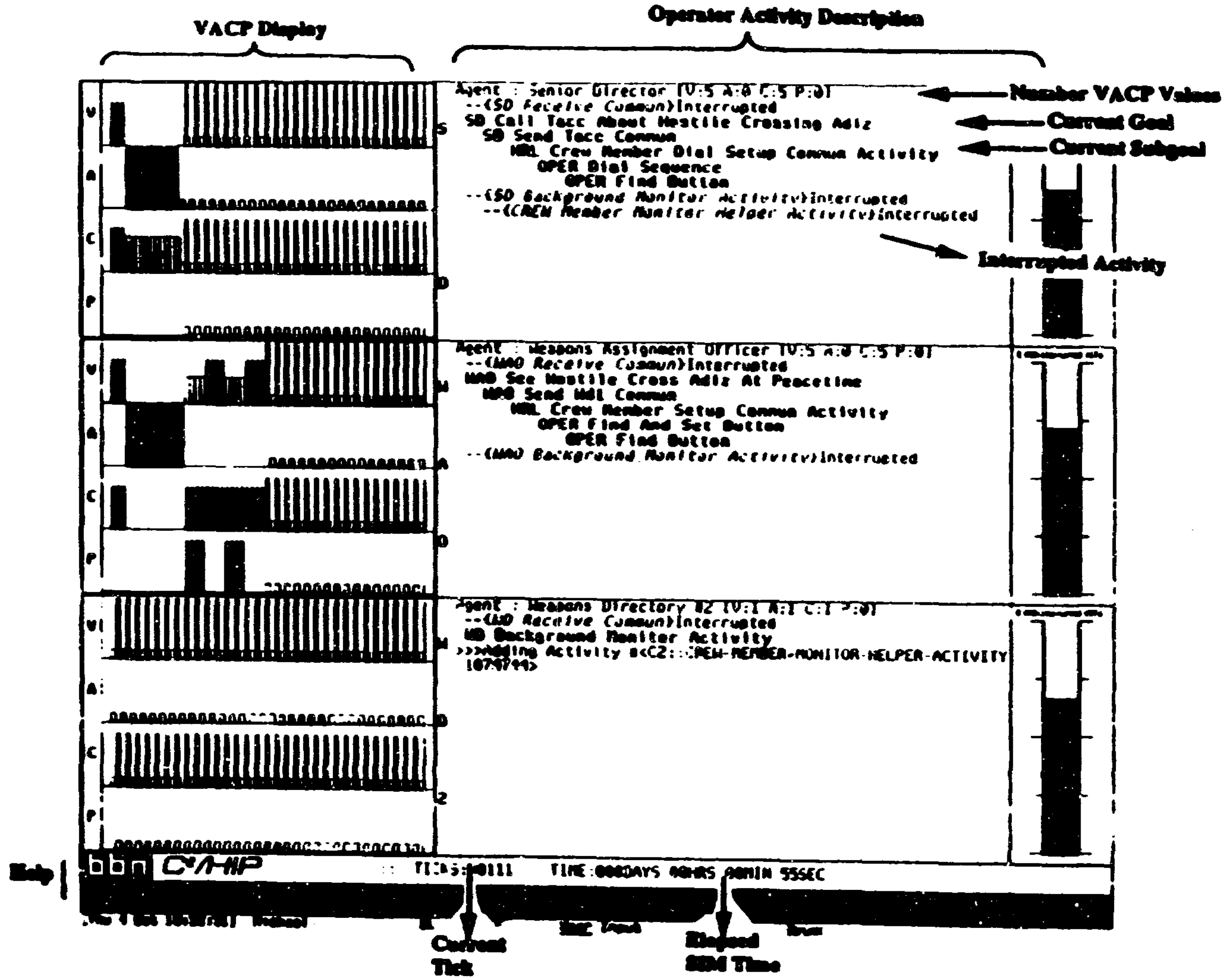


Figure 8. Black-and-White C^2 System Display.

VCAP Display. An animated "strip chart" type of display that shows a set of four bar charts displaying the time dependence of the workload for each emulated crewmember in terms of visual, auditory, cognitive, and psychomotor load.

Operator Activity Description. A textual display showing the current configuration of the rules governing the behavior of the crewmember according to the system's current internal model.

Each crewmember model has a Display CONTROLLER governing its output to each of these two displays. In addition to a pair of displays for each of the crewmembers, there is a corresponding pair of display windows for the airbases, aircraft, and TACC (supreme command) used by the simulation. A similar pair of windows is used for system output and display. Each crewmember model's displays are fully independent, so that the analyst/user running the system is able to select from among the various crewmember models that set which models the information the analyst/user wishes to display.

This model of window output/interactions has a number of significant advantages. True portability is enhanced. Because of the inherently hardware-specific nature of windowing systems, displays and window interactions can be the most difficult features of an existing application to port to a new hardware platform. However, as noted above, in the Display CONTROLLER model all knowledge about the nature of and interactions with the platform-specific windows being used by the displays is highly localized and made modular by being encapsulated within the PWINDOW. As a result, in porting the display-related portions of an applications system to a new hardware platform the only portion that needs to be modified is the PWINDOW itself. Moreover, this conversion is a one-time cost; specifically, it need not be done on a per-application basis. Once a platform-specific version of a PWINDOW has been established, it can be reused for future ports of Display-CONTROLLER-based systems to that hardware platform.

All windows in a given system are completely interchangeable. Again, no knowledge about how the display is to be shown is embedded in the platform-specific window. Consequently, rearranging or redistributing displays for a given system is simply a matter of reassigning the physical windows among the appropriate Display CONTROLLERS and the associated PWINDOWS.

Furthermore, this greatly simplifies the resourcing of these windows. Rarely used or complicated types of displays need not have their possibly space-expensive windows created for a single, short-term use. In other words, there need be no more windows created than the maximum number that can be shown at single time, regardless of their use. In fact, it is simple to turn off any or all displays in a given system.

As an example, in the C^2 evaluation system discussed above, the user/analyst can select which set of crewmember model outputs he/she wishes to display. In this model of window interactions, showing the chosen displays becomes simply a matter of distributing the necessary windows among the Display CONTROLLERS for the models whose output is desired.

Having all output to the screen channeled through the various Display CONTROLLERS provides the system with a centralized locus for controlling, manipulating, or eliminating some or all of its output. Indeed, a program outputting through a specific display need not even know whether its output is actually being shown. This has three advantages:

1. A given portion of a system need not be concerned about whether it is providing output (as stated above, if a disabled display is later re-enabled, the Display CONTROLLER is responsible for updating the display appropriately). In the C^2 evaluation displays of crewmember model data, all output from a given crewmember model is passed through a single Display CONTROLLER. This gives the system a useful, simple way of turning off the display from that model, and no model is concerned as to whether its output is actually being shown. Another, somewhat more complicated example involves the MCE radar display. It is possible to turn off all the displays in the AIRT system. However, in the case of the radar it is critically important that the internal state of the display be maintained (even if the display is not currently being drawn) so that the display will be correct and up-to-date if the displays are re-enabled. Again, the Display CONTROLLER that handles the radar is responsible for maintaining this internal state. In short, it does this each time it is notified of a change, by first making the appropriate change to its internal state and then deciding whether to actually make changes to the display that reflect this change.

2. In certain applications (e.g., complicated, graphics-intensive simulations) where a significant portion of run-time often is devoted to graphics and textual output, the Display CONTROLLER model gives a single, central location for disabling all output to a display when this is desired. In the C^2 evaluation system, with its many complicated displays, it is often desirable, when attempting to run until a specific predetermined time-step, to disable all displays until the run is over. Suppressing the displays for these intermediate steps can allow a great improvement in speed.

3. Highlighting for emphasis is a generally desirable feature of a system. Multiple highlighting, such that when a single entity in the system is selected or singled out to be noticed all representations of that entity currently on the screen become highlighted, is also useful. Under the Display CONTROLLER paradigm, highlighting is simply another aspect of the details of a particular display handled by a Display CONTROLLER. When an entity in the system is told that it needs to highlight its representations on the display screen, it notifies the Display CONTROLLERs responsible for the displays in which the entity occurs, and multiple highlighting is simplified.

Finally, CDIS contains a number of packages for handling specific types of displays. Among those used in the AIRT/HIP system are the BGRID system, which handles the display of realistic-looking, interactive buttons (see Figure 9) and the RIB displays for drawing the paper-ribbon-like displays used for displaying the crewmembers task loading in the analyst's view. (See Figure 8.)

BRAHMS System

The set of software packages and utilities that go by the name BRAHMS (Behavior Representation and Human Modelling Simulation) supports those features of the system that supply the creation, simulation, and monitoring of the human modelling objects, from which it takes its name. The most important of these features are of three classes: human model bases, human performance displays, and executive controller.

Human Model Bases. There are those features that support the human modelling itself. They involve two aspects. First, the software structures that go into the specification of those features of the human model itself that are characteristic of the whole class of BRAHMS-based human modelling (as opposed to those features specific to the application at hand). The types of activities representative of general human operator behavior are as follows:

1. There is a set of behaviors having to do with the selection of what procedure to perform next, given a stack of available procedures. The selection process involves determining the priority of a given procedure (which is determined in a priority matrix for each operator) and examining the resources required to perform that procedure.

2. There is a set of procedures associated with the resource loading model for the operator. Activities are selected on the basis of priority (as mentioned above) and the

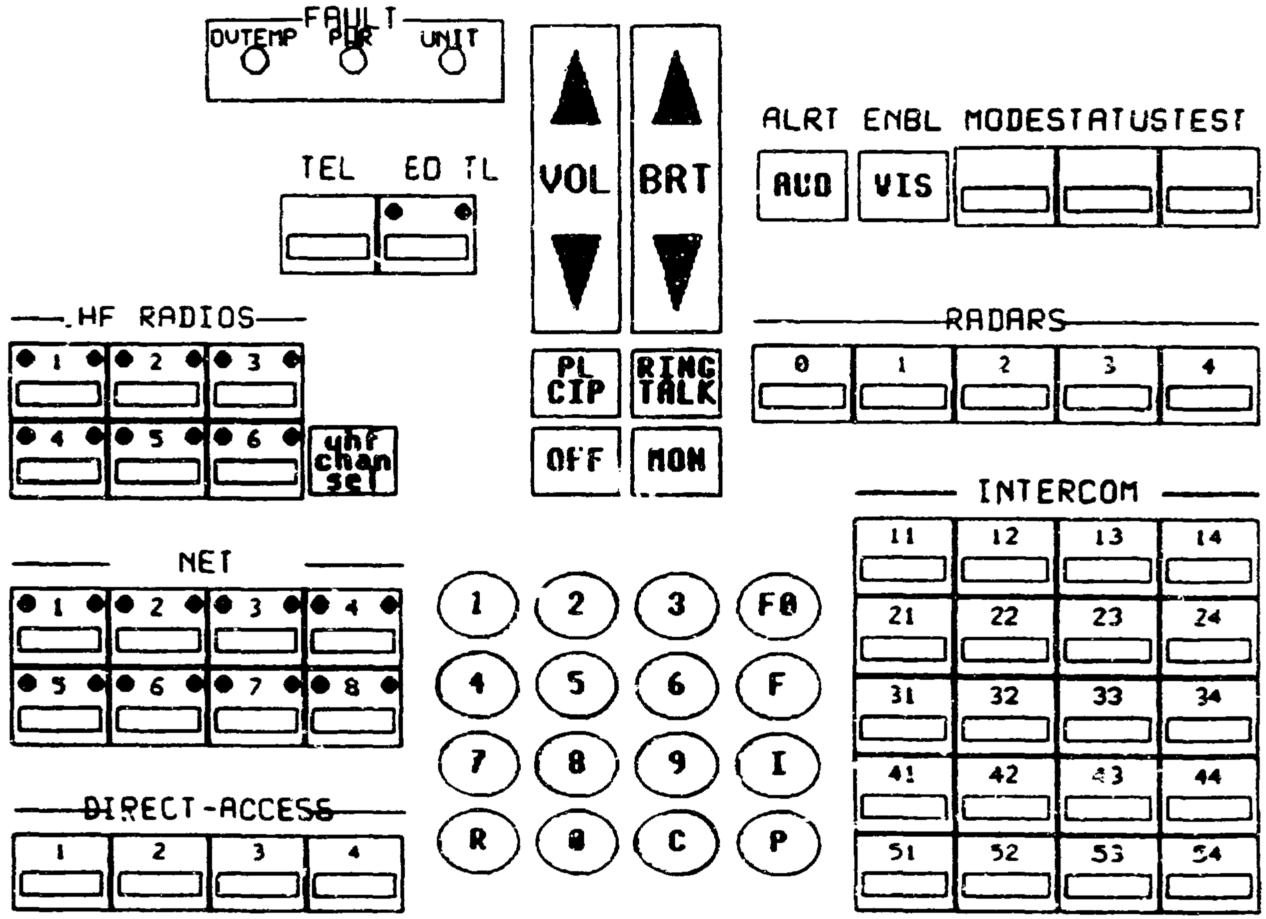


Figure 9. Example of Use of the BGRID System to Display Interactive Buttons.

resources that the operator can bring to bear on the task. This resource model is described in the section of this report dealing with human models.

3. There is a set of procedures dealing with the interruption of one activity by another of a higher priority. Procedures can be interruptable or non-interruptable. In general, if a procedure is performed through a sequence of subactivities, it is able to be interrupted. If interrupted, there are several reentrance and restart options associated with the particular activity being performed. Second, there is a cluster of software structures that specify how the BRAHMS-based activities themselves are structured; these structures have to do with features of the activities that support the above-mentioned interruption features and determine how communication among activities is handled.

Human Performance Displays. BRAHMS supports a set of standard displays whose basic underlying structure is common to all BRAHMS systems (although the specific details corresponding to a specific system may vary). These displays primarily involve the analyst's view and associated system displays.

Executive Controller. Finally, BRAHMS contains a basic, underlying structure used in the BRAHMS-based systems which is centered around the application system's MASTER object. The main components involved are as follows: the basic MASTER object, a basic structure for the analyst's view displays, a basic structure for the AGENT-TOP-LEVEL object (whose responsibilities involve the creation and control of the various active agents corresponding to the application system), and a basic structure for displays used specifically by the application system. In each of these cases, BRAHMS supplies a basic, foundation stratum on which the application system builds, filling in the details as appropriate to its needs.

AIRT and HIP Systems

Each BRAHMS-based system has a set of top-level components specific to the particular application. The AIRT and HIP systems also have common supporting features. Thus, for the AIRT and HIP systems there are: (a) those parts that are specific to the AIRT system, (b) those that are specific to the HIP system, and (c) those underlying features which are common to both, by their relation to Command and Control processes.

We will consider the common features first. Because of the similarity of the AIRT and HIP systems, there is naturally a great deal of overlap in their features. This common set of features constitutes the Command and Control module. The following features are the most important within the C^2 module:

1. A simulation of the real-world situation (i.e., aircraft). In this case, the aircraft behave as independent agents driven by a simulation handled by the system's AGENT-TOP-LEVEL object.

2. The MCE emulation system and its displays. This package draws heavily on the features supplied by the CDIS graphics package described above. It supplies a realistic, interactive simulation of the MCE/radar system and interacts with the real-world simulation, displaying the current state of the aircraft and other components of the world representation.

3. The human model/agents that are common to both the HIP and the AIRT systems.

4. The activities that support these common agents.

5. A set of tools that allows the interactive specification and initiation of a script by the programmer/analyst.

6. A set of tools that can be used to collect and reduce data describing the behavior and performance of the human models.

Also common to the AIRT and HIP systems are a number of additional features which are built on top of those of the C^2 module. Within AIRT, these primarily involve activities and rules that support the higher-level behavior of the air traffic control features of AIRT. The most important of these for the HIP system are (a) a number of modifications to the MCE displays that enhance its interactive nature, thereby enabling it to be used by an operator in a way that simulates the use of an actual MCE workstation; and (b) a set of components that serves as an interface between the operator sitting in front of the MCE simulation and the rest of the underlying HIP system. Because of the object-oriented structure of the system, it is possible to build this interface object as just another object which replaces one of the human models in the underlying AIRT system; this replacement is then, to a great extent, transparent to the underlying system. In a certain sense, the interface object acts as a "wrapper" around the human operator and serves as his/her interface to the rest of the system.

We will now describe how the fundamental components of the C^2 evaluation methodology are linked to create the evaluation system. The discussion to follow concentrates on the function and control flow through the system. Included in this description are references to rules and models (for both human and equipment performance). These components are critical to the analytic behavior of the system. We will develop these key concepts fully in Section IV.

Function Flow

The basic data flow for the system is illustrated in Figure 10. The system is implemented in a modular and object-oriented framework, as previously discussed. The module boundaries in Figure 10 correspond to the conceptual design and implementation distinctions in the system.

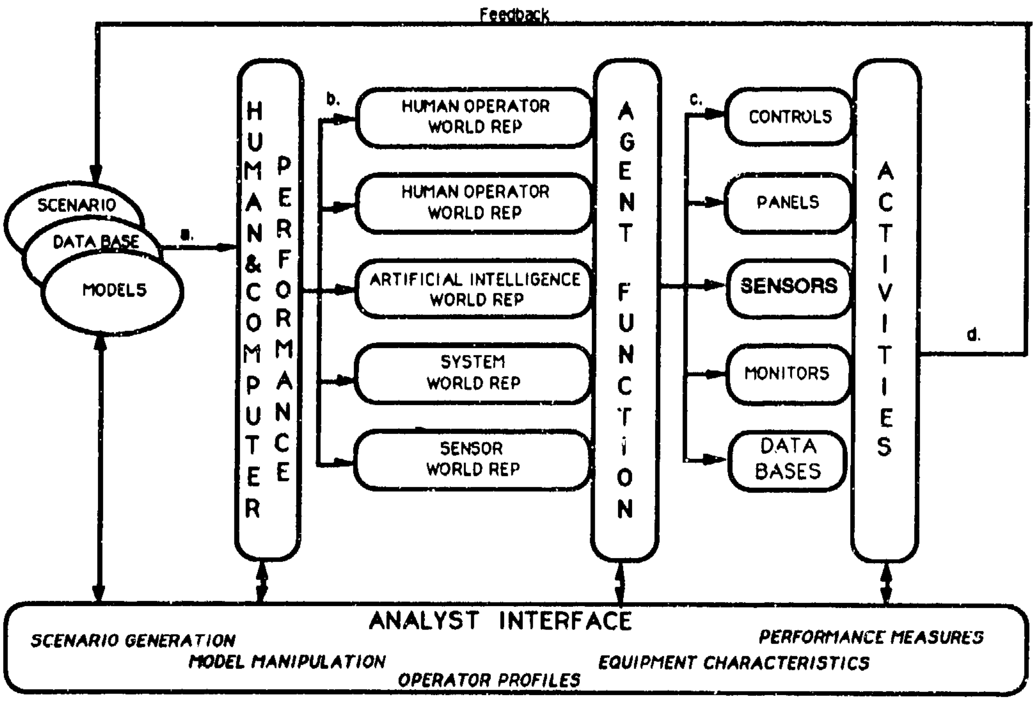


Figure 10. **Basic Architecture for Human/System Performance Analysis Simulation.**

The functional flow through the implementation begins with models of the scenario and environment in which the evaluation methodology is to be exercised. In the case of the MCE automation impact evaluation, this scenario includes geographic and geopolitical boundaries in support of an air defense operation. Also included in this description are object-based representations of Friendly and Enemy aircraft, radar sites, CAP points, and air bases. The activity of the scenario is played out though the emulation of MCE equipment in the MCE operator console. (This equipment is illustrated in the equipment description ovals attached to activities.) The scenario is interpreted through the human operator performance models (Link ***a***). (These will be described in detail in the next section.) The output of these models provides data that modify the world representation of the operators that have interacted with the displays (Link ***b***).

The agent function module is composed of operator-independent abstractions of the processes by which the operator-objects act on the data contained in their world representation. These abstractions currently include communication protocols, interruption/resumption protocols, task-queue management operations, and decision mechanisms. Once action is decided upon, the way in which this action takes place is mediated by descriptions of the system equipment. In the current instantiation, that equipment is the MCE OCU (Link ***c***).

Finally, activities are initiated that invoke models which describe when, how, for how long, and with what resources the operator responds. The effect of these activities is then fed back to reflect changes in the scenario state as a result of operator action (Link ***d***).

For example, as the MCE RGDU displays the appearance of aircraft (directed by the scenario script), the human visual performance models direct the position and dwell time of, for instance, the WAO's gaze as the WAO searches over the MCE OCU. The information (MCE object status) taken in by the WAO is used to update his/her world representation. The state of objects in the world representation is arranged by categories, and attached to these categories are rules of behavior. In our example, these are the set of categories for the WAO under the current Rules of Engagement and Air Tasking Order (ATO). To continue the example: If the WAO's visual scan encounters a Friendly symbol as it passes over the RGDU, a set of rules attached to Friendly aircraft is run to see if the condition of that aircraft meets the criteria for any rules to be activated, or "fired." Attached to Friendly aircraft objects are rules regarding their combat mission state (e.g., paired, enroute, engaged, on CAP) and actions to be taken by the WAO based on time since last observation. These rules can either cause action to be initiated or simply cause the WAO's world representation to be updated. The WAO's rules generally dictate that, given a particular condition of the air battle, communications be initiated either within the MCE (e.g., to direct a WD to communicate with a pilot or alter the current pairings and assignments) or with an external agency (e.g., to scramble fighters to CAP or to an engagement).

The firing of the appropriate rule for action causes an activity to be created (spawned). In turn, that action may have several supporting actions that must be taken to satisfy the termination conditions for that state of action. An activity (e.g., initiate communications) invokes models that describe procedural sequences (what must be done), communications protocols (how it must be done), and motor response requirements (what are the physical parameters for its completion).

Other human operator agents within the MCE are guided by similar perceptual models, but the rules and the activities spawned by those rules depend on the duties and the profile of that operator. So, to continue the above example: A call from the WAO to a WD results in an auditory input to the WD. The WD responds to this change in world state by applying rules to the content of the communication that spawn activities on the WD's part. For example, a request from the WAO to re-pair a fighter will result in the requested re-pairing, and in a new condition (a previously paired fighter now unpaired). The WD object must determine, according to mission state and Rules of Engagement, what is to be done with that fighter. Rules for pairing geometry are invoked which spawn action. Finally, action is taken through the procedures required by the MCE equipment suite.

Functional Manipulation

In addition to these modules, the system provides a set of interface tools designed to facilitate screen-based, mouse-activated manipulation of the objects that comprise the evaluation system scenario. As described below, these tools can be applied at run time (access to scenario conditions or changes in model parameters), or at system definition (flex rule browser, or equipment definition).

The analyst is provided with on-line access to the scenario conditions, to the parameters that describe human performance, to equipment functions and arrangement, and to the rules guiding the behavior of the human agents in the simulation. Examples of these utilities are as follows:

1. One such utility is a screen-based facility for setting up a scenario script. This capability is illustrated in Figure 11. Pathways for Enemy aircraft are established by

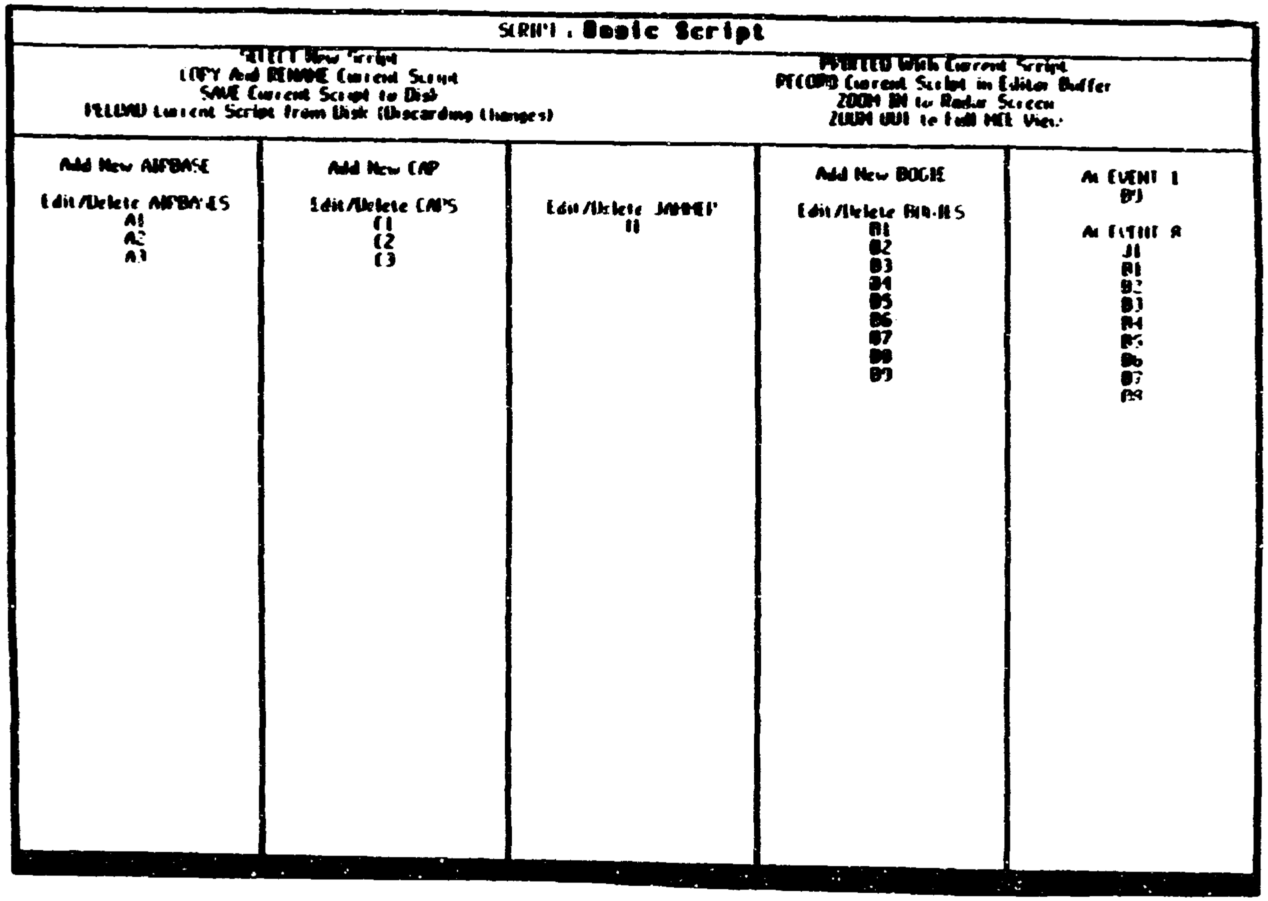


Figure 11. Script Selection and Editing Window.

mouse-clicks. Aircraft type and characteristics are also accessible to the designer. Likewise, airbases, CAPS and scenario events can also be edited.

2. Another provides access to the FLEX rule system. The FLEX rule system treats rules as packets of relations between the individual operators in the MCE evaluation system and the objects in the world representation about which those operators must make decisions. This rule system is illustrated in the matrix of operators and entities in Figure 12. Using a mouse-to click on the entries of the matrix calls up a browser window in which the rules associated with the operator-entity pair are displayed, as illustrated in Figure 13.

The rules are expressed in standard English provided by the programmer and in a first-approximation to English generated from the underlying code by the system. Portions of a rule are highlighted when the cursor moves over them. These portions are the substructure of the rule. The system is currently implemented as a browser. Development of an editing capability from the browser structure is straightforward. The editing system would provide analysts a tool by which the underlying decision rules of operation could be changed.

3. A number of utilities are available for defining the physical and functional characteristics of the MCE equipment. Following the object-oriented implementation paradigm, all of the equipment models can be independently manipulated. Each of the control panels and all of the buttons, switches, and data on those panels are objects. The equipment suite consists of objects that are composed to emulate the MCE equipment. Parameters of these objects specify their size, shape, location, and function. For the case of data representing underlying buttons and switches, a graphics editing interface has been provided to select and position them. All other parameters related to equipment must be entered as text into the LISP structure defining the equipment.

4. In addition, the system has a set of tools to write operational variables to files for performance analysis. This experimental mode of operation will be discussed in that part of Section V which deals with verification and validation.

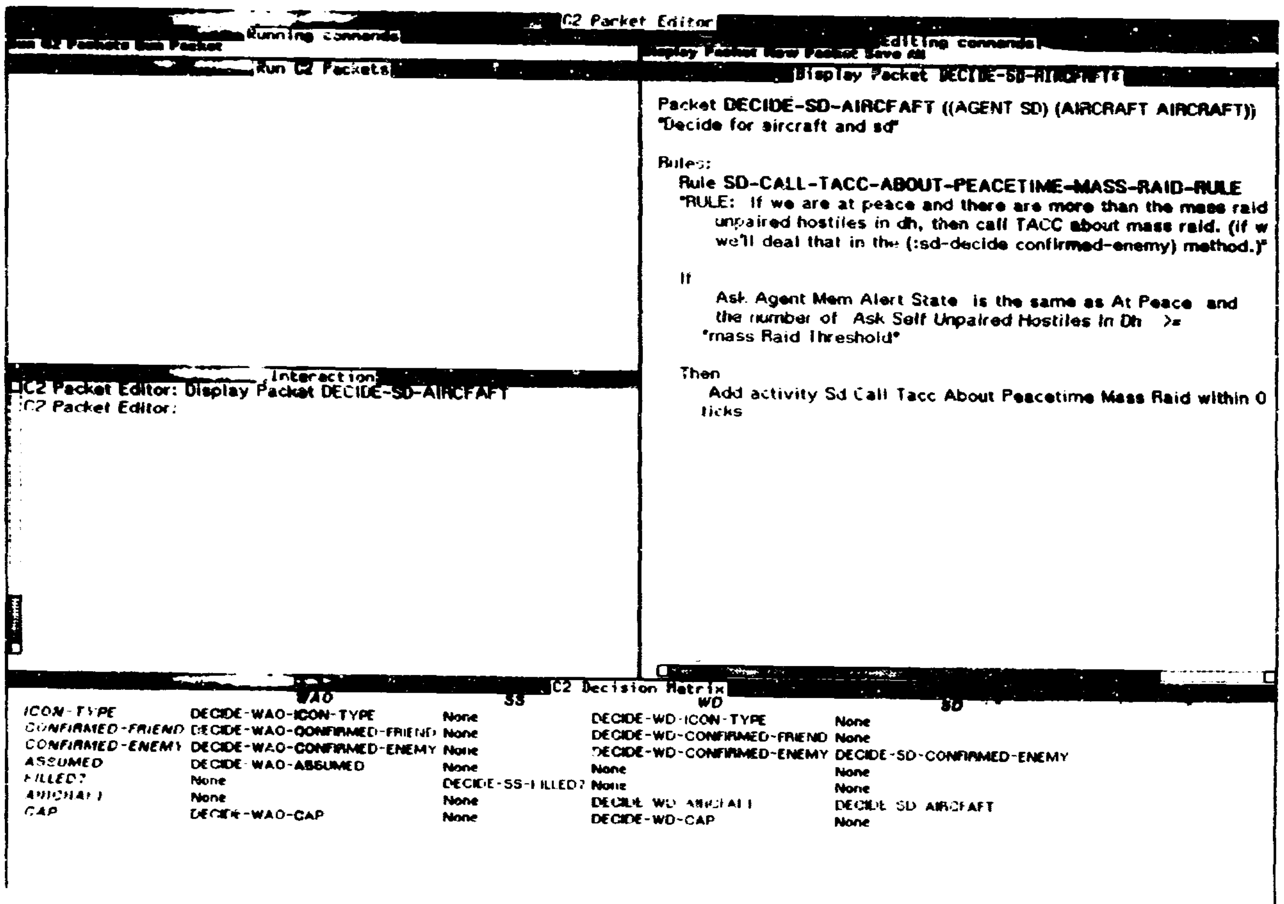


Figure 12. Flex Rule System Editing Screen.

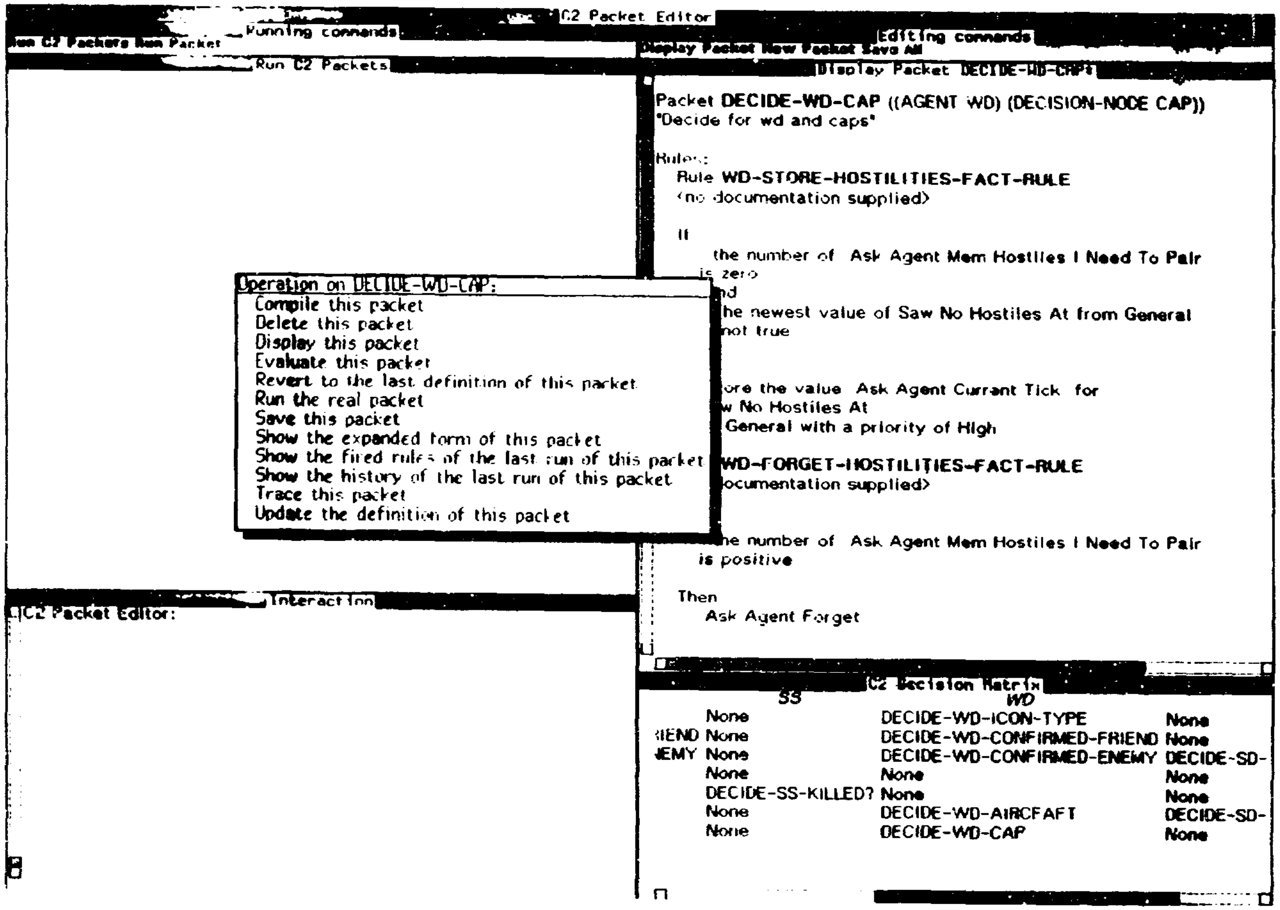


Figure 13. Flex Rule System Editing Screen Showing Browser Window.

Control Flow

The basic architecture for LISP control flow for the MCE is presented in Figure 14 from the point-of-view of the management of information display. The simulation module is essentially the forcing function for the flow of activity in the analysis. Events occur at particular scheduled simulation times (e.g., a particular simulation "tick"), or as an analyst-selected "asynchronous events" invoked with screen-based commands. In this way the simulation module serves as a stimulus to the operator models.

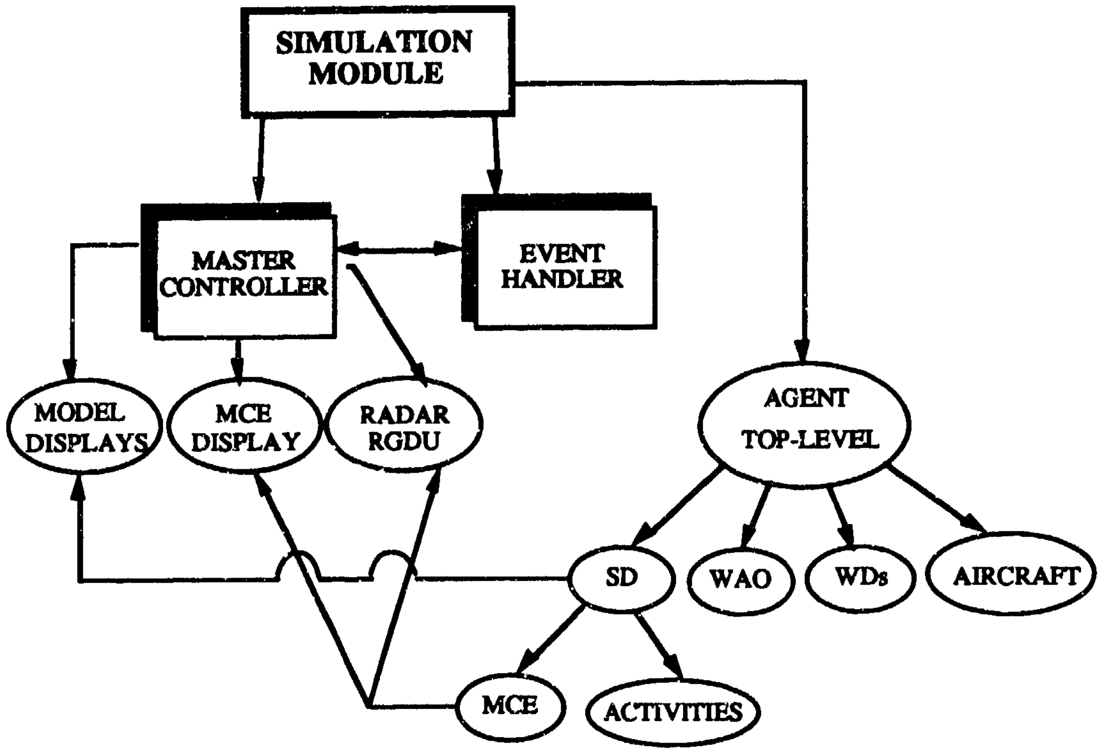


Figure 14. Control Flow for the MCE.

The major control modules are a Master Controller and an Event Handler. The next level of control is found in the operation of the Model Displays, MCE Display, Radar RGDU Display, and AGENT-TOP-LEVEL object modules. Below these are the individual models, the actions of the simulation agents, the function of the MCE equipment, and activities of the radar screen. We will now describe the operation of these modules in some detail.

Master Controller/Event Handler

The Master Controller acts as the system executive and routes information and messages among the system modules. The Master Controller is linked to the Event Handler, which contains two types of events that drive the operation of the simulation objects. "Tick-based" events form the basic script of the simulation and depend on the initial configuration of the agents including Bogie/Friendly aircraft and the Rules of Engagement. The other event type is "asynchronous," which provides for the injection of user-defined events into the operation of the simulation; it also provides a mechanism for conditional events to be defined in the simulation. Event-streams through the Master Controller drive the displays, the radar, and the activities of the agents.

In addition to the display modules for models and equipment, each agent contains an object pre-presentation of the MCE. This means that the world representation of each operator contains the current perceived configuration of his/her AUXPANEL control settings, hooked data readout presentation, and workload parameter settings (VCAU). Because of the graphics-intensive and changeable nature of the radar RGDU presentation, there is one common representation of the air picture that is accessible to all operators.

Operators can configure their screens to different resolutions and to offsets. However, to capture these individual aspects of the radar presentation, we can block an individual operator from the full-scale RGDU by directing the visual attention model to attend only to that portion of the screen currently available to the operator. The radar is represented only in the radar display object and is referred to by the agents through the Master Controller.

Agents

Contained in each crewmember agent is a model of that crewmember's behavior (encapsulated in his/her current set of activities) and his/her MCE console. Associated with each crewmember's activity state is a set of displays. Figure 15 illustrates the model for the MCE equipment associated with the WAO and the model for the radar common to all crewmembers.

As mentioned above, each crewmember agent contains an internal representation of his/her own MCE console. Internally, the states of all the buttons, etc. are recorded and maintained. A view of only one crewmember agent's MCE is shown at any one time. At that time, the MCE console displays of all the other agents are disabled, without affecting

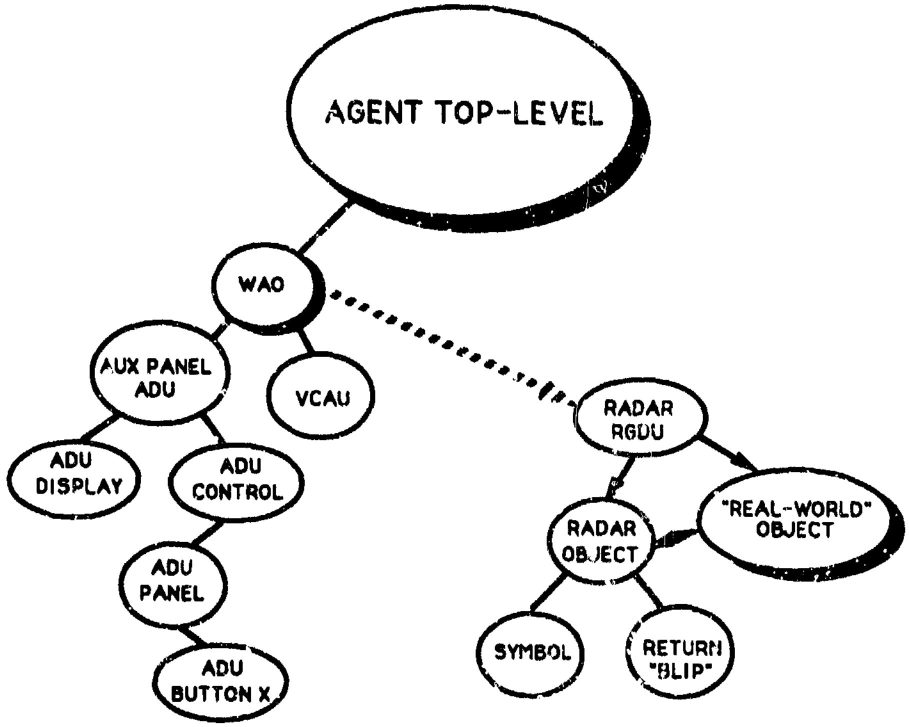


Figure 15. Agent Structure.

in any way the internal representation of the MCE's state (for more on this point, see the discussion of the Display CONTROLLERs). The sole exception to this is the representation of the radar and its display; at present, a single radar object is held in common and used by all the agents' internal representations of their MCE consoles.

An agent in the C^2 system is used to represent an independent, free-standing entity capable of and responsible for initiating its own behavior. This behavior is controlled by the set of activities that the agent is currently running. These activities are spawned in response to changes in the agent's environment in the system.

Activities

An activity is a unit of behavior governing the action of an agent. Roughly, it is a piece of code that runs for a short duration, until a given goal is accomplished or until the goal is otherwise terminated. During its lifetime it will, at various times, execute code to affect the behavior of its parent agent or send messages to other agents in the system or to features of the simulation.

The structure of an activity can be recursively hierarchical; that is, it can itself spawn subactivities when it needs to delegate subtasks that must be performed. For example, a crewmember agent might need to communicate with another crewmember, as a response to the appearance of an unknown object on the radar screen. To accomplish this task, the crewmember agent would spawn a high-level "communicate with crewmember" activity. This activity would have small subactivities such as "dialing" up the other crewmember, talking to the other crewmember, and "hanging up" the communication. Each of these subactivities would, in turn, have many subtasks (reaching for and pushing buttons, looking to verify that a button-click "took," etc.).

An example of a complicated, high-level agent with activities is that of the crewmembers in the C^2 system. These agents, as models of human behavior, gain information about their environment as described by the system by means of their auditory and visual models. The human model/agents then respond to the changes in their resultant internal model of the world by modifying the set of activities that the agents have running at that moment.

The internal representation of the world of these agents with activities is also governed by these agents' auditory and visual models. At given intervals, the agent looks at or listens to its environment and detects and collects information about the world, which it stores in its memory. The C^2 agents scan their equipment or look at particular buttons and switches according to the activities they are carrying out. If the agents are not currently active, they scan their console as a background task. The interval of this background scan is one glance at a new section of their equipment every 2 seconds. In addition, the operator agents are constantly listening to or monitoring the radio channel to which they are connected. According to what they then perceive about the world, the agents respond to the world by modifying the set of activities that is running.

Aircraft objects provide an example of a less complex type of agent in the C^2 system. These objects also respond to changes in their environment by spawning new activities, but the model of their interactions with the rest of the world is much simpler than the models for for the human agents. In short, aircraft objects simply respond to incoming messages sent to them by the human crewmember agents. (For example, the crewmember agent WD1 might send an aircraft agent a command to return to an airbase for refueling.) Basically these agents merely receive and respond to direct communications from the crewmember agents.

Governing all the agents is an entity known as the AGENT-TOP-LEVEL. The AGENT-TOP-LEVEL is responsible for keeping track of and handling communications among the various agent objects and the other entities of the system. It is also responsible for various system maintenance "housekeeping" tasks such as making sure all the agents are activated at the appropriate moment.

Activities are encoded as LISP procedures and methods that describe what is to be done, what are the enabling conditions for the performance, who takes the action, the action's duration and load, how the action is successfully completed, and how the activity is terminated or interrupted.

Each tick-step for a given agent is divided into three parts or passes:

1. *Pre-Tick.* In this pass, the agent decides which of its current set of activities it will run on this tick. This decision can be very simple. For instance, in the aircraft object agents, all the available activities are in a strict linear order of precedence, and the currently available activity with the highest priority gets to run.

Alternatively, for the human agents that depend on the visual, auditory, cognitive, and psychomotor (VACP) load models, each agent must first sort his/her current set of activities according to a pre-determined priority. Next, this set is then reviewed in order, and each top-level or parent activity is asked to decide if the conditions are appropriate for it to run on this tick? If not, this activity is skipped. If the activity can be run, its VACP for this tick is calculated and the corresponding total loads for the agent are incremented. If one of the four V, A, C, or P loads becomes too great, this activity cannot be run on this tick. This process continues until all the activities are processed or all the loads are filled.

2. *Tick.* In this pass, the actual work of the activity is done, messages are sent to other entities, etc.

3. *Post-Tick.* In this pass, some side-effects of tick are cleaned up. For instance, if the activity spawned a new sub-activity during the tick pass, the new activity is actually queued-up and spawned during the post-tick phase. (This is done to avoid a cascade effect, to prevent an agent that receives its tick after the current agent from getting a resulting message one tick out of phase with the current agent.)

For example, in the activities governing communications, the agent who has initiated the communication has a "send communication" activity, and the agent to whom the communication was sent has a "receive communication" activity. The structure of individual communication is illustrated in Figure 16, which depicts a typical communication between the WAO agent and the WD1 agent. In this case, the WAO is the initiator of the communication and the WD1 is the recipient of that communication.

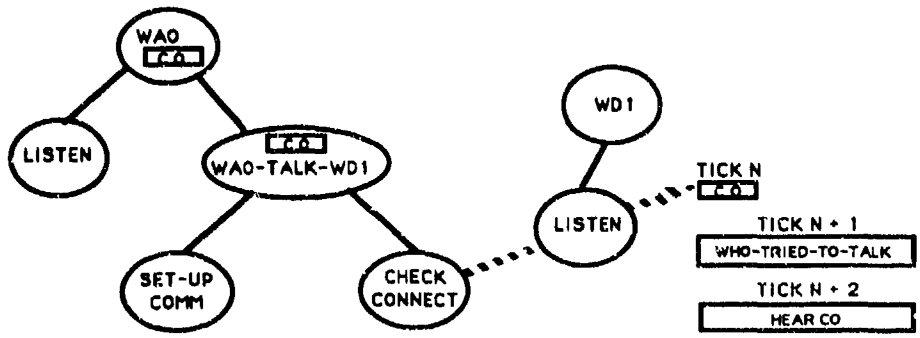


Figure 16. Communication.

During the pre-tick pass for sending, the agent, after first determining that no activity of higher priority is pending, must decide if it is still appropriate for the "send communication" activity to run. For example, the sending agent checks to see if the receiving agent is still connected, or if the receiver has been interrupted by activities of his/her own with higher priority than listening to the communication. If the receiver is still available, the message is sent. During the tick pass, the receiving agent determines who is trying to communicate and whether that communication is of sufficiently high priority to be heard. In the post-tick pass, the receiving agent actually hears the content of the message.

The simulation issue addressed in this multi-pass paradigm is that activities or communications on the part of one agent may change the world situation and context of action on the part of another agent on the team. Queueing and prioritization are required, as well as a period in which to allow decisions to settle into the new context which each tick of activities brings to the situation.

IV. REPRESENTATION STRUCTURES AND MODELS FOR OBJECTS AND AGENTS

The representational formalisms established in the C^2 methodology are a key feature to the generality and power of the methodology. The analytic methodology must have some "ground truth" that holds the state conditional information about the simulation objects and with which each of the intelligent agents in the simulation will interact.

That ground truth will be of several types. First, there are the characteristics of the mission or operation being performed. In this domain of discourse, C^2 GCI, an adequate and universally accessible description of the airspace in which the aircraft (Friend and Foe) are operating is required. This description includes terrain and geopolitical features, as well as cultural information such as cities, air bases, and targets. In addition, tactical military features such as CAP points, safe corridors, missile engagement zones (MEZs) and air defense intercept zones (ADIZs) must be represented. The second kind of ground truth information required is that associated with operating procedures. Significant items of this type include the air tasking order and appropriately focused fragments thereof, intercept preferences and geometries, and manning procedures. In addition to these fairly high-level procedural concerns, there are within the CRC procedures having to do with the local command structure and standard operating procedures. Finally, there is information that deals with the behavior of aircraft in the physical world (velocities, accelerations, fuel usage, missile envelopes, etc). In providing the ground truth for representation in simulation, we have used three representational formalisms appropriate to each of the three information types (declarative, procedural, and physical/dynamical).

Declarative Information

Declarative information is organized by taxonomies based on whole-part relationships. For example, an area of responsibility (AOR) for a given WD is part of an area of operations under the control of a given CRC. That area of operations is part of a theater of operations under the control of the TACS, etc. That taxonomic relation extends to descriptions of equipment in the MCE.

For example, the operation control module (OCM) is composed of four primary objects: the radar graphics display unit (RGDU), the Voice Communications Access Unit (VCAU), the auxiliary control panel, and the auxiliary display unit (ADU). These objects are in turn composed of buttons, switches, and display surfaces. In addition to the representation of the physical condition of the OCM modules, each operator carries internal representation of the state and condition of his/her individual control station.

It is of some significance that, though ideally the human operators' representation of the world is consonant with the state of the simulated world, our system makes no assumption that this is the case. The internal representation of the human operator may be differently structured than the ground truth, it may contain more data or less data than the basic representation of declarative knowledge. This capability for both systematic and random deviation from the ground truth of the simulation world is a critically necessary component of any system that intends to represent and analyze significant human performance. Further discussion of the ramifications of internal representation of the human operators modelled by this system is deferred to the discussion of models in the following section.

Procedural Information

Procedural information (i.e., how something is done) is held and structured by a goal/task procedure taxonomy. The activities performed by human agents are formulated as human performance models (visual search, decision-making, memory functions, etc.) These models are called into action as a procedure is invoked that requires prediction of a human operator's performance in response to mission/operational requirements. The operation of this methodology is critically dependent on the accuracy and validity of these models. We will describe each model operation in detail after this overview.

The taxonomic structure of actions to be accomplished places them in a mutually satisfactory relationship. For example, the high-level goal, to conduct satisfactory air defense operations, is served by manning CAPs and managing Friendly resources. These, in turn, are served by a subgoal, to conduct appropriate pairings between Enemy and Friendly forces, which in turn is satisfied by appropriate activation of tasks for pairing, which, in turn, is constrained by proper operation of the MCE equipment at the OCM. This goal/task procedure hierarchy is further structured by the notion of priority in the tasks being performed, and by reasoning about the capacity of the equipment and about the capabilities and limitations of the operators interacting with that equipment.

Physical Relations

The physical relations in the world representation are maintained and calculated by dynamic equations of motion (limited in this implementation to simple point-mass equations) and models of aircraft expenditures over operating time. The motions of aircraft are first-derivative indications of velocity over the screen space (pixels per tick) and represent a velocity of 1.2 times the Enemy aircraft velocity. Expenditure of fuel is calculated as a relation between speed modes (cruise, pursuit, and afterburner) and expert opinion as to how long fighter aircraft can function at those speeds.

Representation System Benefits

The value of the infrastructure that organizes the C^2 methodology is seen in the flexibility and tractability of the system to accommodate design and procedural changes while maintaining its capability to provide performance predictions for the human/machine system. In addition to the robust response of the analytic system to procedural, declarative, and physical changes to the entities modelled in the analysis structure, several other benefits accrue as a function of the organized and integrated information structure in the C^2 methodology.

As noted by Card, Moran & Newell, (1983) in their discussion of cognitive architectures, the definition and implementation of a processing structure is theoretically and pragmatically useful in that the structure provides a framework and constraints on the degrees of freedom associated with the individual models of cognitive phenomena used to describe human/machine functions. We will elaborate the models in the next section.

Further, the structure of the system provides a common vocabulary with which to discuss the relative contributions of human operators and intelligent automation. The parameters of a given performance can be assigned as a function of assumed automation or calculated as a function of human performance models.

Finally, the structure of procedures and the representation of the human agents in the system make explicit the basis for human performance in response to the scenario. Once validated, this cognitive structure will allow a systems analyst to examine, without ambiguity, the rules and knowledge base whereby the human operator is taking action.

Human Performance Models

Human performance models are the basis for the C^2 evaluation methodology's prediction of the MCE crew's operation in the face of demands of the tactical air defense scenario. Our general approach to modelling is the perspective that the human performs critical information processing operations and control functions in the C^2 environment. This perspective asserts that "human performance varies because of differences in the knowledge that a person or team of people possess (both the form and the content), in the activation of that knowledge and in the expression or use of that knowledge" (Woods, Roth, Hanes & Embrey, 1986, p. 6).

This perspective (one of several useful views of human performance) meets our needs in describing human operation in a complex information processing and communication system. To provide a relatively complete and useful representation of the human operator in these systems, we need to account for three aspects of the operator's behavior: perceptual processes, cognitive processes, and response processes. Specifically, we assert that the human operator models must provide the following model descriptions:

1. A computational description of human visual processes that describes both general visual scanning and directed visual processing.

2. A description of scanning patterns and dwell times for information processing, as well as predictions of what is and what is not available to the visual system for inclusion in a knowledge cache for further processing.

3. A description of audition that includes a description of the effect of monitoring multiple channels simultaneously. In addition, the model of human communication must include a mechanism for interruption of messages, as well as an assignment of priority to the incoming information.

4. A description of interactive models representing cognitive processes. This includes a description of the state of the operator's knowledge of the world (an updateable world representation); a method that describes the process of decision-making based on rules, as well as on more heuristic and algorithmic calculations; a model of the function of memory (both working memory and long-term information store); and a model of how cognitive activity might influence perceptual processes (specifically, how problem-solving and planning might direct vision to seek information from the OCM displays and controls, and direct communication functions to ask for the required information from other operators, pilots, and personnel up the chain of command).

5. A description of how the human operator, having made a decision or plan to initiate activity, goes about effecting that activity within the constraints of MCE operation. This model should describe both human neuromotor response and verbal communication protocols.

General Model Design Issues

We will describe our implementation of each of the above sets of models directly. There are, however, several general issues in human modelling formalisms that should be introduced. These are the use of normative versus. descriptive behavioral models, the levels of detail of the models, the mathematical foundations of the models, and finally, the integration principles for the models. We will describe each issue and then address how it has been resolved in this methodology.

Normative versus Descriptive Models Issue. The distinction here is whether the model describes the human operator's behavior as it "ought" to be, according to a normative set of assumptions (normative), or whether the model simply describes the human operator's behavior based on some set of data collected for that behavior (descriptive). Examples may help to clarify the distinction. In the normative case, we might model human decision-making in terms of the Bayesian principles of conditional probability (Hayes, 1973). It is known that human operators generally do not make optimum use of conditional information; so, this model, while providing a general outline of human information processing, would be considered normative in the sense that it describes how an ideal decision maker would use conditional information. In the descriptive case, we might model human decision-making based on actual research. For example, in modelling the human's response following presentation of a stimulus, the response time to be used in the model could be derived from the mean reaction time response of many operators tested on a similar task. This would be a descriptive model in the sense that it is based on the data from empirical tests, as opposed to a theoretical formal foundation.

In the C^2 evaluation methodology, we have favored the inclusion of normative rather than descriptive models. This was done in large part because of the prototype nature of the equipment. There simply are insufficient data about human operators' performance on which to base a descriptive model. Also, the normative modelling approach is more in keeping with our goal to create a generally useful analysis tool that is not tied to any one configuration or equipment set.

Levels of Detail Issue. The granularity or level of detail issue concerns how much detail is necessary to provide a sufficient description of the behaviors of interest.

In the C^2 evaluation methodology, we have provided models of performance at levels of detail adequate to describe the *observable* effects of operator action as the simulation is executed. Visual scanning of the radar screens, for instance, determines what the operator sees. Timing and positional accuracies are critical to this behavior and are modelled at a very fine level of detail (ocular position variations of 1 degree of visual angle subtended and temporal variations of 200 msec are calculated). Alternatively, human decisions are modelled on the basis of rules and heuristics. We do not, in this instance, calculate decision times, as these times are likely to be small in comparison to the other operational inertias in the system. Insofar as we have been able to determine, this architecture is unique in its ability to accommodate the interaction of models at varying levels of granularity (Elkind, Card, Hochberg & Huey, 1989).

Probabilistic versus Deterministic Issue. The mathematical assumptions under which models are developed determine the predictive power and applicability of those models in a given domain. Deterministic models are sufficient to describe the operator's behavior in open-loop control systems. In the deterministic model development, the operator models will respond in the same way at every presentation of the same scenario. The same aircraft will be assigned to the same bogies, etc. The value of the deterministic approach is that the analyst can pursue what-if analyses with the assurance that the only changes in the methodologies output will be a function of his/her manipulations. On the other hand, probabilistic models provide options and variances in operator behavior described by distributions of the probability that an action will be taken, or that a signal will be perceived. These probabilities are enacted by relating the signal/noise and prior probabilities of the stimuli to the filter and plant characteristics of the human operator. Cognitive processes such as situation assessment may be deterministically represented in a rule base or described probabilistically using Bayesian or evidential reasoning techniques. Memory processes can, similarly, be described in terms of probabilities of recall, or deterministically described using queueing theoretic models.

With respect to the probabilistic versus deterministic issue, we have in this implementation restricted ourselves to deterministic models. We chose this alternative, in conjunction with AFHRL, to rule out probability-based causes from our assessment of automation impacts on performance. At a fine level of analysis, the deterministic nature of the models allows us to pinpoint the system changes to be examined without the requirement to provide a multiple-run statistical basis for effects. If in the future probabilistic models are introduced into the analysis methodology, the architecture can support multiple-run, or Monte Carlo, methods of operation.

It is of importance to note that operation in the HIP mode (i.e., with human operators interacting with the simulated operators in a scenario) immediately moves the methodology into a closed-loop mode of interaction in which the effect of earlier performance is used to guide later performance. Human operators are not deterministic in their response. Further, human operators are very sensitive to the temporal resolution of a system's response and issues of dynamic system stability must be addressed.

Model Integration Issue. As discussed earlier , our evaluation methodology uses a modularized object-oriented paradigm for human/system representation. The human performance models follow this paradigm as well. Models describing individual perceptual, motor, and cognitive processes are encoded as objects and methods on those objects. Communication among models (representing the processes of perception, cognition, and action) is provided through LISP-based message-passing protocols. The action of these models is the sole basis for operator-object knowledge in the simulation. There is no higher or meta-level repository of simulation knowledge and very few global representations of the operator's process.[3]

Concerning the issue of model integration, we have designed the system as a modular framework rather than a monolithic structure. A fully integrated and monolithic "mega-model" of human performance does not seem appropriately responsive to the analyst's needs to investigate in more or less detail the operations of the C^2 structure in response to automation. The more modular structure also frees the analyst from the often constraining assumptions inherent in monolithic models.

Human Performance Model Implementation

In this section we will elaborate on the human performance models currently implemented in the C^2 evaluation methodology. We feel this implementation is sufficient to provide insight into the impact of automation on CRC-level C^2 activities. This set of models should by no means be construed as a complete description of C^2 activity in general; however, extension of the methodology's capabilities is made easier by the modular structure of performance prediction. Each of the operators modelled in the C^2 evaluation system is an active and independent agent. These operators take action based on the state of their internal representation of the C^2 GCI world.

Agent World Representation. A human's cognitive representation of the world is a complex structure, the characterization of which has been the topic of intense research

---

[3]There is currently no consensus among practitioners as to the "correct" integration approach for these models (see Chubb et al., 1988 for a discussion). The rationale for our approach is provided in Corker et al., 1989). We will elaborate on this integration with MCE GCI examples after our discussion of the individual models that have been implemented.

interest by experimental and cognitive psychologists. (See Collins & Smith, 1988, or Baron & Corker, 1989, for a review of these issues as applied to complex, dynamic human/machine integration.) Though we are sensitive to the fact that the state of knowledge in cognitive science continues to evolve, we assert that the following structures and methods are necessary and provide testable hypotheses for human performance. We have implemented these structures within the AIRT and HIP systems.

The operators modeled in this system interact with the MCE through perceptual processes and activities. Each operator has an individually-defined world representation. This representation contains (a) a declarative description of the world as the operator knows it, (b) a set of actions and procedures that are within the operator's capacity to perform, and (c) a set of rules that guide the application of these actions. The operator-object builds this representation from a standard base of assumed "knowledge" (e.g., the C^2 operators know what aircraft are and what their characteristics are with regard to counter-air-defense operations). The operator-object assumes that information about a particular scenario will be provided verbally and "heard," or presented visually on the MCE equipment and "seen." As discussed, all such transactions take place through message-passing protocols among objects.

The declarative world knowledge is represented in a taxonomy in which the higher-level objects (e.g., the MCE OCM) are composed of lower-level objects (e.g., the radar display graphics unit, the auxiliary control panel, etc.). As previously described, procedures are arranged in a goal/subgoal/task procedure taxonomy, with increasingly specific actions needed to satisfy the accomplishment conditions of the procedures. Finally, the rules and knowledge base are structured in a propositional framework that conditionalizes action; for example, "IF an aircraft has been scrambled and has been on CAP for x minutes, THEN call for a fuel check at time y" (where x and y are aircraft-specific time intervals).

A fundamental distinction is made in our system as to whether knowledge about the world is stored as facts (termed above "declarative knowledge") or as actions and relationships (termed "procedural knowledge"). Currently the world representation of the operator's knowledge is predominantly declarative. The objects in the world are represented in a frame-theoretic paradigm. Objects are defined by characteristics called "slots," and these slots are filled with values calculated through the simulation. In this way the knowledge structure is static in its organization, but dynamic and calculable in terms of the particular values assigned to the world at a given point in time through simulation. For instance, an aircraft-object in the simulation is defined by its altitude, velocity, bearing, expendables, mission, call sign, etc.; the particular values of that altitude, bearing, etc. are determined by that aircraft's action in the simulation. Aircraft behavior can be determined by the analyst (e.g., Enemy aircraft ingress routes) or be based on responses to simulation states (e.g., Friendly aircraft response to Enemy tactics).

The operator-object similarly represents the aircraft (once it is "seen" through the action of the visual perception model) as an object with slot structures that correspond to the original object. Ideally, the perceptual state of the operator mirrors the state of the real ("simulated") world. As noted previously, what is intriguing about this structure is that operator-objects can have internal models that differ from the simulated world both in terms of the slot values they assign to their internal representation and, in a more complex way, in terms of the structures they assign to comprise their world objects.

The internal organization of the operator has two artificial bits of information that are not strictly perceptually available but that serve simulation purposes. These are an identification attached to a bit of information as to its source (i.e., the piece of equipment from which it was derived or its auditory source), and a temporal tag identifying when the information was received. These are used to identify where and when information was received (or not received) to support a post hoc analysis of operator behavior. In the case of the temporal tag, the information is also used to spawn anticipated tasks (e.g., "check fuel x minutes from time y when latest aircraft state was available").

Memory Models. The memory of an operator-object is defined in terms of the declarative and procedural taxonomies that structure the operator's internal representation. We adhere to the distinction between active or short term memory and long-term memory stores (Klatsky, 1984). In the operational environment we are modelling, long term memory holds the *a priori* knowledge with which the operator-objects enter the simulation. We model active memory as a limited queue of procedures and a limited set of declarative information. (The current limit of these queues is 10 items or procedures, in keeping with the generally defined limits of working memory.)

Information (declarative information or actions that need to be taken) is forgotten through three parallel mechanisms. First, there is the fundamental queue-length limitation. If the operator-object has more than 10 procedures or more than 10 declarative bits of information active at the same time, new information replaces the oldest information from the queue in a "first-in-first-out" (FIFO) regime. Second, the entry of information into the active memory queue is time-tagged. Items are forgotten according to an exponential decay function (Peterson & Peterson, 1959,). (See Corker, 1990a, for details.) Third, the activity level of memory access is accounted for by an inverse ratio between the number of memory accesses and the permanence in the memory store. The more the operator-object is forced to use the active memory store for input or retrieval, the less likely older stores will be accessible, again following a FIFO queueing discipline.

We are aware that such a simplistic structure does not fully capture the complexity of human memory processes. In particular, it does not address semantic relatedness or valence/priority of the information held in memory, nor does it address empirically determined effects such as primacy (first things in active memory tend to be remembered better than middle items). Even given these constraints, the structure does force the issue of time-bounded utility in information management by human operators.

Visual Processing Model. The operators of MCE equipment must constantly scan their equipment to keep their mental images of the radar air picture information updated. To account for the time and movement required to find and fix target data in the MCE operator console visual field, we have implemented a model of visual scanning. Each of the activities in the mission simulation script has an attribute which identifies what equipment (and what sequence of interaction) is required to respond to mission demands. The majority of visual attention in MCE operation is required to be foveal (e.g., reading data, making bearing and range estimates, locating and operating control panel switches). Foveal vision covers only a small part of the entire visual field. (The region defined as foveal is .5 degree of visual angle, whereas peripheral vision approaches 180 degrees of visual angle, Graham, 1965.) There will be two sorts of visual scanning that inform the internal representation of the operators according to the following mechanisms.

The first, "active gaze," represents the focused and directed movement from the current point of regard to a target point. The action is characteristic of cases in which the to-be-attended object is in a known position. The motion is a straight line from the present position to the target.[4] The operator is assumed to be 18 inches from the center of

---

[4]Though there may be a contribution to motion through head movement, we will not consider that at this time.

the OCU. The velocity of ocular motion is 100 degrees per second. There is a 200-millisecond pause between eye motions (i.e., saccades). The visual scan will cover all the displays of the OCU. The specific parameters that describe this model's operation (e.g., the distance of the operator from the screen, the speed of ocular motion, and the dwell or pause time) are variable slots in the model's definition. In this case and in all other model definitions, we have attempted to instantiate the best data available regarding human performance to guide model operation. However, in every case, we have made the variables that define model function manipulable by the analyst to facilitate exploration of alternative functionality.

The second type is a monitoring or search pattern. The saccades in such a search pattern typically last for 50 milliseconds and cover about 10 degrees. Again, there is a 200-millisecond pause between movements. The effective radius of a fixation in this scan is about 14 degrees from the center of fixation (Stark, Vossius & Young, 1962).

Visual scanning also is directed by the decision-making and problem-solving tasks of the simulated operator. We have implemented the following visual dynamics into the simulation:

Visual Attendance. When a "viewable" referent is named (heard or spoken), thought about, or otherwise entered into a human being's attention, he/she tends to fixate the referent immediately. This tendency is more or less independent of whether the person seeks or requires information from the referent. However, when information-seeking or interpretation is not involved, such fixations may be brief.

The simulated operator will look at the referent to which it is attending. For example, when listening to a communication about a particular plane, the operator will shift its gaze to that aircraft; when thinking about the need to call a pilot about one thing or another, the operator will fixate the image of the relevant aircraft; and when describing a pairing to a pilot, the operator will look at the image of the plane with which it is communicating, the bogey about which it is communicating, and the planned intersection point that it is communicating. These fixations will be coordinated with the verbal mention of the referents, thus constraining the speed of running the whole pattern (Carpenter & Just, 1976; Cooper, 1974; Kahneman, 1973).

When no relevant visual object is available for fixation (or when the referent is available but displaying distracting characteristics), human viewers tend to direct their visual gaze to some non-informative and thus non-interfering locus, such as some empty point in the air between them and the screen (or any other visible surface) as long as they are concentrating on the related thought. When the simulated operator has nothing to do and the screen has been relatively static, that operator will look at any new object that appears on the screen or that is moving or blinking. This may also be a reasonable heuristic for initiating operator attention to a developing scenario. In the simulation of the operator's mind and memory, such observed objects will be registered so that when the commander mentions them, they (and any other obvious or inferable characteristics) will already be represented in location in the operator's mind.

Spatial Problem Solving. Eye movements provide insight into spatial problem-solving and, moreover, tend to mediate the process. When the simulated operator is thinking about interception points and optimal pairings, its eye movements should mimic its thoughts. In our evaluation methodology, we have modelled behavior such that for *each* pairing considered, the operator's gaze will fixate on the target aircraft, the candidate Friendly, and the projected point of intersection between them. If the operator must consider more than one pairing at a time to allocate Friendlies to bogies properly, fixations on all such triads of locations will be included in the decision process. When the operator has settled on a pairing or set of pairings, its (their) components should be fixated again to reflect the decision and record the decision in memory.

When people view a moving object, they tend to compute the object's projected path and to use it in any subsequent visual search for the object. The research literature does not permit parametric estimates of the robustness of this ability across time or intervening cognitive events. However, it can be expected to degrade in several different ways: (a) The greater the elapsed time since the operator last attended to a moving object, the greater will be the x-y-z error in estimated location that results from imperfect estimates of the object's velocity; (b) variance in estimated time since the operator last attended to a moving object can only increase with the amount of time that has elapsed; (c) the greater the number of intervening events since the operator last attended to an object, the more difficult it will be for the operator to recall what he/she last knew about (Carpenter & Just, 1976; Gould, 1976; Russo & Rosen, 1975).

In our system, if the simulated operator must return its gaze to a moving object (or if it must reestablish its location after killing a jammer), it will generally begin its search at the location where the object is projected to be, rather than at the location at which the object was last attended. This projection is based on linear extrapolation, with fixed positional variance based on a Gaussian distribution.

Rule-Based Decision Models. The operator-objects make decisions about what actions to take, generally guiding communication with each other or with the aircraft they are controlling or assignment of Friendly assets to Enemy targets (pairing) based on rules and algorithmic calculation of the value of a particular pairing. The rules are structured as propositional statements that are peculiar to each operator's area of responsibility. An example of such a rule packet is shown in Figure 17.

Rules:

WAO-CONFIRMED-ENEMY

"Rule: IF we see a confirmed-Enemy who is in Friendly territory
AND [we haven't seen it before OR when we saw it before it wasn't
a confirmed-Enemy] THEN ... IF alert-state isn't WAR then call for
VID, otherwise wait time x and check to make sure it's paired."

If C^2 Is In Friendly Territory And Buffer and
The newest value of Icon Type from Ac Object is not Jammer and
The newest value of Seen Killed? from Ac Object is not true

Then IF Ask Agent Mem Alert State is the same as At WAR THEN
Add activity Wao Check Pairing of Hostile Over Border within Wao Wait
Time Until Pairing Enemy ticks
ELSE Store the value Ac Object for Hostile over Border on Pending Vid
with a priority of High and
Add activity Wao See Hostile Cross Adiz At Peacetime within 0 ticks

WAO-CONFIRMED-ENEMY-PAIRING

"Rule: If we're at war we want to hook&look at the pairings the
furthest into Friendly territory."

If The newest value of Alert State from General is the same as At War and
Current Time - Ask Agent Mem Time Since Last Random Hook >
* WAO Ticks Between Random Hook and Look and
Paired Fighters and
Ac Object is one of Paired Fighters

Then Add activity Wao Low Level Hook Activity within 0 ticks with Ac
To Hook
First being
First of Paired Fighters
with Ac To Hook Second being Second of Paired Fighters

Figure 17. Examples of Decision Rules.

The calculation of pairing values is provided as a linear-weighted combination of several factors, as follows (Henry, 1989):

1. Time to Intercept

2. Distance of Intercept from Home CAP

3. Current Friendly Fighter Heading

4. Whether Friendly Fighter is Paired

5. Target Heading

6. Whether Target is Paired to Another Friendly Fighter

The rule-based decision model we have used is fairly stiff (i.e., not adaptable to unanticipated changes in the air space situation.) It is also based on single-point observations of the world; that is, patterns of Enemy action are not anticipated or recognized. Further, this model assumes there is no uncertainty in the incoming data and no ambiguity as to which rules to fire in response to these data. Clearly, the domain of human decision-making, particularly decision-making under stress, is critical to the success of an expansion of this methodology to consider resource allocation and battle management. However, the behaviors exhibited by the operator-objects in the MCE has been examined by experts in the field and found to be adequate and representative.

Further, these behaviors are the subject of our ongoing verification and validation effort. The details of this work are contained in the next section.

V. CONCLUSIONS

This section on the conclusions reached during our work in the design, development, and implementation of the C^2 evaluation methodology is divided into three sections. The first is a section on the work of the verification and validation stage of our contract. The second is a lessons-learned review that includes both implementation and conceptual issues in this type of modelling for evaluation purposes. The last section summarizes the possibilities and the requirements for further research on modelling the complex interaction among human/machine systems in the area of C^2 evaluation.

Verification and Validation

The C^2 evaluation methodology system provides an environment in which the impact of automation on tactical C^2 operations can be tested. In the provision of this capability, several aspects of the C^2 operational environment were modelled. Each of these elements (i.e., human performance models, operational algorithms and rules, equipment representation, and warfare area representation) needs to be verified and validated to some level of acceptability. The level of verification and validation required is, of necessity, an evolving process -- one of moving toward robust verification of human model performance. Initial evaluation is needed, however, to ensure a firm basis for more extensive testing. We feel that before going forward with extensive analysis of tactical C^2 and expanding this methodology into broader areas of C^2 operation, a careful evaluation of the efficacy of the modelling approach is required. Toward that end, BBN and the Air Force Human Resources Laboratory have designed an aggressive and extensive verification and validation plan.

In addition to an examination of initial face validity, validation and verification of the C^2 evaluation methodology can be divided into three areas: Operational Validation and Verification (V&V), Automation and Rapid Prototyping V&V, and Human Performance V&V. Within these areas there are three levels at which performance can be evaluated:

1. A high-level or procedural testing phase using identical and somewhat simple scenarios (i.e., involving a single WAO and two WDs handling a small number of Enemy sorties), as will be detailed in the following sections;

2. A medium level of testing in which the requirements for the defensive strategy are more complex and in which there is concern for the logistics of airbase resources and aircraft turnaround time;

3. A detailed level of testing in which the human performance models of resource management, decision-making, and planning are examined in a complex air defense scenario (Henry, 1989).

Face Validity

In terms of face validity, judgments were provided by experts in the areas of tactical C^2 operations and MCE training, throughout the contract development effort. (An acknowledgment of these experts is provided in the Preface of this document.) The feedback of experts throughout the development of the C^2 evaluation methodology has provided guidance that has enhanced the face validity of the final system. Similarly, the rules of behavior that guide the simulated operators' behavior have been incrementally examined and improved to enhance face validity. The conclusions of our expert judges have been that the "look and feel" of the C^2 evaluation system are sufficient to support integration of human operators into the system, as discussed in the section of this document describing the HIP operation. An excellent test of the face validity of the modelled MCE performance will be provided in follow-on tests in which the human operator interacts with the simulated operators and the simulated system in operational tests.

Operational Validity

A more rigorous test of the system's performance has in part been completed under this contract. This was a test of the procedural and operational validity of the modelled system and operators in simple simulations of an air defense scenario.

These tests were performed at BBN and at the USAF School of Aerospace Medicine facility at Brooks Air Force Base, Texas. Basically, the tests called for the C^2 evaluation methodology, with the current operator models, to be tested against human operators handling the same mission demands.

A regime of counter-air defense scenarios was run, and data on dependent operational variables were collected. The scenarios were developed by the research staff and C^2 subject-matter experts at AFHRL, Wright-Patterson Air Force Base, Ohio. The scenarios involved three configurations of Enemy aircraft: (a) single pair of bogies, (b) dual pairs of bogies, and (c) a wave of four bogies. These configurations were run through six attack patterns with increasing degrees of complexity -- from a straight-on attack, through flanking and weaving, to a column pattern with a turn to CAP. A full description of the scenarios and the coordinates for their operation is available in Appendix B, "V & V Data." The operational variables that were collected for the AIRT response to attack were as follows:

1. Total number of hostile tracks killed.

2. Duration of hostile flight prior to kill.

3. Distance hostile travelled until kill.

4. Perpendicular of hostile distance north of the ADIZ at kill.

5. Total of hostiles killed outside the ADIZ.

6. Total number of Friendly fighters used.

7. Frequency/duration of communications among elements.

The data for these tests are available in Appendix B.

The C^2 evaluation system in the AIRT mode of operation, as described previously, is a deterministic system in terms of the models and rules that guide behavior. This allowed us to run the system through the scenarios and collect the data in a single pass. The human operators' performance on the other hand, as should be expected, is characterized by some variance; therefore, statistical procedures must be applied to these data for comparison to the AIRT process. USAF AFHRL is currently performing that statistical reduction on their data. Therefore, as of publication, we are unable to provide any conclusions as to the outcome of the study.

Lessons Learned

There have been a number of observations and insights gained from the experience of the C^2 evaluation methodology development. These fall into two categories: system development lessons and human performance modelling lessons.

System Development Lessons

One observation is that the modular nature of the object-oriented coding process played an essential part in the successful development of system representation. The modularity of the code allowed us to develop the various evaluation system components in parallel. For instance, the representation of the rules of operator behavior could proceed at the same time as the representation of the physical devices of the MCE. This modular development allowed us to take advantage of subject-matter expertise as it became available rather than adhering to a sequential system development.

Second, the system was developed through a layered architecture that attempts to isolate specific implementation of the MCE from system elements of a more general nature such as the rules and models for human performance. This allowed us to respond to system upgrades in terms of the MCE functionality without having to reimplement the human performance models.

Third, the transition from a fully automated and model-based implementation of the AIRT system to the HIP implementation has both positive and negative lessons-learned. On the positive side, the modular interface between the interface management systems in AIRT and the human performance models that use those interfaces has expedited the inclusion of actual human performance into that analysis system. The late decision to attempt to include human performance has caused some reconsideration of the infrastructure of the AIRT system. For instance, there is no formal way to pass such information as ROEs out to a human operator. Also, there is no self-reflection process whereby the modelled operators can communicate their world view to the human operator. Such information is critical to successful team operation. For example, such a simple output as "No, I don't see him" in response to a query is not supported in the current versions of AIRT.

In the future, designers of the evaluation system should explicitly consider what information in the system is needed for full communication with the world and which is legitimately within the system and can be passed "under the table" in the LISP code. This will be especially important in the expansion of the analysis system into battle management operations.

Human Performance Modelling Lessons Learned

Our experience with providing an integrated structure for the interplay of human performance models has been an illuminating one. We believe that we have provided a unique environment for model interaction, as we have discussed in an earlier section on human modelling. However, there are some significant developments needed in human performance modelling to support the evaluation system's further development and validation. We will discuss areas where models are inadequate, or are entirely missing, to support the analysis.

Communications. We believe that there are serious limitations in the current model of human communication. The model assumes several unlikely conditions to be true. First, it assumes all communications to be of the same length. Second, the mechanism for interruption assumes that once the message has begun to be delivered, it will continue to be delivered without interruption. Third, the bandwidth limitation of a message being heard or not (i.e., depending on how many other channels are being monitored at the same time) is artificial. We feel that this model can be readily improved through consideration of the type of information being passed and a better model of bandwidth limitations.

Resource Limitations. The current resource model is static and based on subjective opinion as to the task difficulty according to visual, auditory, cognitive, and psychomotor loads. This approach is an adequate place-holder for an important phenomenon. However, given the limited sources of estimation for the subjective load, and the static nature of their use (i.e., loads do not interact across modalities, nor do they change over time), we feel this representation is inadequate. There have been some advancements since this work was initiated and some models of task load as a dynamic phenomenon are being developed. These models should be investigated as to their applicability in our domain.

Decision-Making Limitations. The human operator decision system is contained in a set of rules and algorithms that relate the state of the world and the state of the operator's knowledge about that world to the activation of behavior. This is a reasonable structure and we have recently implemented a rule browser to make this structure visible to the analyst. The deficiency, as we see it, is that the basis of application of the rules is too limited. Currently, the system makes decisions on the basis of a single event or a single message. To capture the complexity of upper echelon tactical behavior, that single-event basis for decision will need to be expanded to include patterns of behavior, the accumulation of evidence, and a decision-weighting scheme (e.g., evidential reasoning, multi-attribute decision-making, or weighted rule applications).

Memory. We believe we have a uniquely developed capability to expand and improve models of human memory in high information load conditions. Using the taxonomic structure of world representation, we can implement selective short- and long-term memory strategies. The structure to investigate such phenomena as attention-narrowing under stress or short-term memory overload is available. We will need to expand this work to identify the methods for memory manipulation to exploit its utility.

Future Research

The required work for future developments in the C^2 evaluation methodology should consist of implementation developments and model developments.

In implementation, we feel the next important step is to make the code and the system more portable. To do this we suggest moving the code from its current FLAVORS implementation to the Common LISP Operating System (CLOS). To make the window system equally portable, we suggest moving the window system to the Common LISP Interface Manager (CLIM). Further, we feel that the system can be easily expanded to interface with real USAF C^2 equipment and software, or be used to investigate the impact of GCI in the network wargaming environment.

In terms of model development, the inadequacies of the current model implementation should be addressed by expanding the current models and/or developing new models for the C^2 evaluation environment. This model development should include the rigorous "low-level" verification and validation effort previously described. Probably the greatest impact on C^2 system analysis will come from the development of improved situation-based models of decision-making and from implementation of the improved memory models. The development of resource and communications models is also of obvious benefit, but they constitute fewer of the operations undertaken at upper echelons.

VI. REFERENCES

Baron, S. & Corker, K. (1989). Engineering-based approaches to human performance modeling. In McMillan, Beevis, Salas, Strub, Sutton, & Breda (Eds.), Applications of Human Performance Models to System Design. New York, NY: Plenum Press, (pp. 203-218).

Birtwhistle, G. M., Dahl, O. J., Myhraug, B., & Nygaard, K. (1973). Simula Begin. Philadelphia, PA: Auerbach.

Card, S. K., Moran, T. P., Newell, A. (1983). The Psychology of Human-Computer Interaction. Hillsdale, NJ, Lawrence Erlbaum Associates.

Carpenter, P. A., & Just, M. A. (1976). Linguistic influences in picture scanning. In R. A. Monty & J. W. Senders (Eds.), Eye movements and psychological processes. Hillsdale, NJ: Erlbaum.

Chubb, G. P., Laughery, K. R., & Pritsker, A. B. (1987). Simulating manned systems. In G. Salvendy (Ed.), Handbook of Human Factors. New York: John Wiley & Sons, pp. 1298-1327.

Collins, A., & Smith, E. (1988). Readings in cognitive science. San Mateo: Morgan Kaufmann Publishers.

Cooper, R. M. (1974). The control of eye fixation by meaning of spoken language. A new methodology for the real-time investigation of speech perception, memory and language processing. Cognitive Psychology, 6, (pp. 84-107).

Corker, K. M. (1990a). Operational concept document for the methodology for evaluation of automation impacts on tactical command and control C^2 systems: human-in-process software. (Unpublished manuscript). Cambridge, MA.

Corker, K. M. (1990b). Software user's manual, for evaluation of automation impacts on tactical command and control C^2 systems: human-in-process software. (Unpublished manuscript). Cambridge, MA.

Elkind, J. I., Card, S., Hochberg, J., & Huey, B. (1989). Human performance models for computer-aided engineering. Washington, DC: National Academy Press.

Goldberg A., & Robson, D. (1983). *SMALLTALK-80*. Menlo Park, CA: Addison-Wesley.

Gould, J. D. (1976). Looking at pictures. In R. A. Monty & J. W. Senders (Eds.), Eye movements and psychological processes. Hillsdale, NJ: Erlbaum.

Graham, C. (Ed.) (1965). Vision and visual perception. New York: John Wiley & Sons.

Hayes, W. (1973) Statistics for the social science. New York: Holt, Rinehardt & Winston.

Henry, E.H. (1989). Verification and Validation White Paper, and personal communication.

Kahneman, D. (1973). Attention and effort. Englewood Cliffs, NJ: Prentice-Hall.

Klatsky, R. L. (1984). Memory and awareness: An information processing perspective. New York: Freeman.

Lane, N. E., Strieb, M. I., Glenn, F. A., & Sherry, R. A. (1981). The Human Operator Simulator: An Overview. In J. Moraal and K. F. Kraiss (Eds.) Manned system design, methods equipment and applications.

Peterson, L. R. & Peterson, M. (1959). Short term retention of individual verbal items. Journal of Experimental Psychology, 58 (pp. 193-198).

Pritsker, A. B. (1984). Introduction to simulation and *SLAM II*. New York: John Wiley & Sons.

Russo, J. E., & Rosen, L. D. (1975). An eye fixation analysis of multialternative choice. Memory and Cognition, 3 (pp. 267-27[illegible]

Schriber, T. J., (1974). *Simulation using GPSS*. New York: John Wiley & Sons.

Stark, L., Vossius, G., & Young, L. (1962). Predictive control of eye-tracking movements. Transactions of human factors electronics, HFE-3 (pp. 52-57).

Woods, D., Roth, E., Hanes, L. F., & Embrey, D. (1986). Models of cognitive behavior in nuclear power plant personnel: A feasibility study. (Report #NUREG-CR-4532) U.S. Nuclear Regulatory Commission.

VII. GLOSSARY

ADIZ	Air Defense Intercept Zone
AFHRL	Air Force Human Resources Laboratory
AI	Air Interdiction
AIRT	Automation Impacts Research Testbed
AOR	Area of Responsibility
ASO	Air Surveillance Officer
ASOC	Air Support Operations Center
ATO	Air Tasking Order
BAI	Battlefield Air Interdiction
BBN	Bolt Beranek & Newman
BRAHMS	Behavior Representation and Human Modelling Systems
CAP	Combat Air Patrol
CAS	Close Air Support
CDIS	Control Display Interface System
CLIM	Common LISP Interface Manager
CLOS	Common LISP Operating
CRC	Control and Reporting Center
CRP	Control and Reporting Post
DCA	Defensive Counter Air
FACP	Forward Air Control Post
FEBA	Forward Edge Battle Area
FIFO	First-in-First out
GACC	Ground Attack Control Capability
GCI	Ground Controlled Intercept
GPSS	General Purpose Simulation System
FLAVORS	The Symbolics LISP Machines Object-Oriented System
HIP	Human-In-Process
HOS	Human Operator Simulator
ID	Identification

LISP	Programming Language that Uses List Structures
MCE	Modular Control Equipment
MEZ	Missile Engagement Zones
OCA	Offensive Counter Air
OCM	Operation Control Module
OCU	
RDGU	Radar Display Graphics Unit
ROE	Rules of Engagement
RTB	Return to Base
SD	Senior Director
SS	Surveillance Supervisor
SSO	Search Scope Operators
SLAM II	Simulation Language for Alternative Modelling
TACC	Tactical Air Control Center
TACS	Tactical Air Control System
USAF	United States Air Force
V & V	Validation & Verification
VACP	Visual, Auditory, Cognitive, and Psychomotor
VCAU	Voice Communications Auxiliary Unit
WAO	Weapons Assignment Officer
WD	Weapons Director

VIII. APPENDIX

APPENDIX A: SYSTEM UTILITIES

* LUDD MISCELLANY: LUDD also contains a number of other commonly used packages and features. The COMMANDER package allows the simple specification of a top-level, keyboard-character and mouse-driven command-loop for the system. For instance, in the HIP system there are two separate such loops. When the analyst's view configuration is displayed, keyboard input is being handled by a COMMANDER loop that handles top-level, system-like commands. When the human-in-the-loop configuration is being shown, a separate COMMANDER loop is in place that handles commands appropriate to that state.

The DATADIR system allows the definition of easy-to-maintain directories of data files that can be assigned to a given applications system.

The SYSLOAD package is a Common LISP compatible software system specification and manipulation tool (i.e., analogous to the Symbolics Defsystem or the UNIX MAKE systems).

The ACT package supplies the basic, underlying structure that supports the activities (and their inference engine and simulation) on which the human models are based. *

* WIREUP-CHILDREN: In this pass, each object is responsible for first creating its children objects and then passing to each of them in turn a WIREUP-CHILDREN message so that they will, in turn, create their children.

*WIREUP-SIBS: In this pass, each object is responsible for gaining the pointers that it needs to any system objects to which it is connected in a non-tree-like manner; that is, to any objects which are not its direct parent- or child-object(s). The present object can gain these new objects either by being passed such objects as arguments to this message or by "reaching back up/down the tree" and sending its parent/child an appropriate message (i.e., whatever is most appropriate for its and/or the system's needs). Once it has connected itself to its own sibling objects, the object then passes the message on to its own children objects.

* WIREUP-FINAL: This pass is used in those cases in which it is not possible to achieve all necessary connections in the above two passes. It is typically not needed.

* INITIALIZE: In this pass the object is responsible for initializing its state (and its children's) to run the system.

To support the creation of the various system objects, a GET-<NAME> function is defined (using the DEFGETTER form). For example, associated with the HIP-MASTER is a GET-HIP-MASTER function. The GET-<NAME> function is responsible for two things.

First , if the object has already been created (i.e., the GET-<NAME> function has been called previously), the GET-<NAME> function simply returns the object.

On the other hand, if this is a regular (i.e., non-master) object and the object has NOT been created, the function creates the object, sends it a "WIREUP- CHILDREN" message (thereby creating all its children-objects), and then returns the object.

If the object is a MASTER-object and it has been created, calling the function simply returns the object. If the object has not been created, this function behaves slightly differently. Besides creating the object and sending it the WIREUP-CHILDREN message, it also sends the object the other creation and initialization messages above. Note that this means if the system has not yet been created, calling the GET-<NAME> function for the system's MASTER object (e.g., the GET-HIP-MASTER function) will cause the MASTER object to be returned with the complete system and all of its component objects in place and initialized.

APPENDIX B: V & V DATA

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: C2; Base:10 -*-

EVENT NAME:  "scenario-1-event-1a"
                      HOSTILE TRACKS KILLED: 1
 HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
                     FRIENDLY FIGHTERS USED: 6
      NUMBER OF SCALE EXPANSION CHANGES: 0
                    NUMBER OF OFFSETS DONE: 0
Data For Killed Tracks, at time of Kill:
 Total Time Traveled   Total Distance Traveled   Distance Above ADIZ
       139.00Sec                016.41NM                    049.63NM
Communication-data
  CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
     TO: TACC  DURATION: 10 TICKS
     TO: TACC  DURATION: 10 TICKS
     TO: WAO  DURATION: 10 TICKS
  CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 8)
     TO: SD  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: AIRBASE-00015  DURATION: 10 TICKS
     TO: AIRBASE-00015  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
  CREWMEMBER: WD1 (TOTAL NUMBER COMMUNS 4)
     TO: AC-00010  DURATION: 10 TICKS
     TO: AC-00010  DURATION: 10 TICKS
     TO: AC-00010  DURATION: 10 TICKS
     TO: AC-00010  DURATION: 10 TICKS


EVENT NAME:  "scenario-1-event-1b"
                      HOSTILE TRACKS KILLED: 1
 HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
                     FRIENDLY FIGHTERS USED: 5
      NUMBER OF SCALE EXPANSION CHANGES: 0
                    NUMBER OF OFFSETS DONE: 0
Data For Killed Tracks, at time of Kill:
 Total Time Traveled   Total Distance Traveled   Distance Above ADIZ
       113.50Sec                013.40NM                    044.92NM
Communication-data
  CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
     TO: TACC  DURATION: 10 TICKS
     TO: TACC  DURATION: 10 TICKS
     TO: WAO  DURATION: 10 TICKS
  CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 8)
     TO: SD  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: AIRBASE-00035  DURATION: 10 TICKS
     TO: AIRBASE-00035  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
  CREWMEMBER: WD2 (TOTAL NUMBER COMMUNS 6)
     TO: AC-00031  DURATION: 10 TICKS
     TO: AC-00031  DURATION: 10 TICKS
     TO: AC-00031  DURATION: 10 TICKS
     TO: AC-00031  DURATION: 10 TICKS
     TO: AC-00031  DURATION: 10 TICKS
     TO: AC-00031  DURATION: 10 TICKS


EVENT NAME:  "scenario-1-event-2a"
                      HOSTILE TRACKS KILLED: 2
 HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
                     FRIENDLY FIGHTERS USED: 6
      NUMBER OF SCALE EXPANSION CHANGES: 0
                    NUMBER OF OFFSETS DONE: 0
Data For Killed Tracks, at time of Kill:
 Total Time Traveled   Total Distance Traveled   Distance Above ADIZ
       158.00Sec                017.56NM                    035.08NM
       180.30Sec                021.31NM                    037.37NM
```

```
Communication-data
  CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
     TO: TACC  DURATION: 10 TICKS
     TO: TACC  DURATION: 10 TICKS
     TO: WAO  DURATION: 10 TICKS
  CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 11)
     TO: SD  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: AIRBASE-00068  DURATION: 10 TICKS
     TO: AIRBASE-00068  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: AIRBASE-00068  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
  CREWMEMBER: WD1 (TOTAL NUMBER COMMUNS 9)
     TO: AC-00063  DURATION: 10 TICKS
     TO: AC-00064  DURATION: 10 TICKS
     TO: AC-00063  DURATION: 10 TICKS
     TO: AC-00064  DURATION: 10 TICKS
     TO: AC-00063  DURATION: 10 TICKS
     TO: AC-00063  DURATION: 10 TICKS
     TO: AC-00064  DURATION: 10 TICKS
     TO: AC-00063  DURATION: 10 TICKS
     TO: AC-00063  DURATION: 10 TICKS


EVENT NAME:  "scenario-1-event-2b"
                     HOSTILE TRACKS KILLED: 2
HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
                   FRIENDLY FIGHTERS USED: 9
    NUMBER OF SCALE EXPANSION CHANGES: 0
                   NUMBER OF OFFSETS DONE: 0
Data For Killed Tracks, at time of Kill:
 Total Time Traveled   Total Distance Traveled   Distance Above ADIZ
       212.00Sec               023.56NM                    040.02NM
       153.50Sec               018.12NM                    035.84NM
Communication-data
  CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
     TO: TACC  DURATION: 10 TICKS
     TO: TACC  DURATION: 10 TICKS
     TO: WAO  DURATION: 10 TICKS
  CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 16)
     TO: SD  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: AIRBASE-00079  DURATION: 10 TICKS
     TO: AIRBASE-00079  DURATION: 10 TICKS
     TO: AIRBASE-00079  DURATION: 10 TICKS
     TO: AIRBASE-00079  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
  CREWMEMBER: WD2 (TOTAL NUMBER COMMUNS 10)
     TO: AC-00078  DURATION: 10 TICKS
     TO: AC-00078  DURATION: 10 TICKS
     TO: AC-00077  DURATION: 10 TICKS
     TO: AC-00078  DURATION: 10 TICKS
     TO: AC-00077  DURATION: 10 TICKS
     TO: AC-00077  DURATION: 10 TICKS
     TO: AC-00078  DURATION: 10 TICKS
     TO: AC-00077  DURATION: 10 TICKS
     TO: AC-00077  DURATION: 10 TICKS
     TO: AC-00077  DURATION: 10 TICKS
```

EVENT NAME: "scenario-1-event-3a"
HOSTILE TRACKS KILLED: 4
HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
FRIENDLY FIGHTERS USED: 7
NUMBER OF SCALE EXPANSION CHANGES: 0
NUMBER OF OFFSETS DONE: 0
Data For Killed Tracks, at time of Kill:

Total Time Traveled	Total Distance Traveled	Distance Above ADIZ
116.50Sec	014.56NM	046.40NM
139.50Sec	017.44NM	057.29NM
234.50Sec	027.68NM	049.89NM
215.00Sec	023.89NM	051.69NM

Communication-data
CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
TO: TACC DURATION: 10 TICKS
TO: TACC DURATION: 10 TICKS
TO: WAO DURATION: 10 TICKS
CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 16)
TO: SD DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
TO: AIRBASE-00101 DURATION: 10 TICKS
TO: AIRBASE-00101 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: AIRBASE-00101 DURATION: 10 TICKS
TO: AIRBASE-00101 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
CREWMEMBER: WD1 (TOTAL NUMBER COMMUNS 13)
TO: AC-00100 DURATION: 10 TICKS
TO: AC-00099 DURATION: 10 TICKS
TO: AC-00100 DURATION: 10 TICKS
TO: AC-00099 DURATION: 10 TICKS
TO: AC-00099 DURATION: 10 TICKS
TO: AC-00099 DURATION: 10 TICKS
TO: AC-00099 DURATION: 10 TICKS
TO: AC-00099 DURATION: 10 TICKS
TO: AC-00099 DURATION: 10 TICKS
TO: AC-00100 DURATION: 10 TICKS
TO: AC-00099 DURATION: 10 TICKS
TO: AC-00100 DURATION: 10 TICKS
TO: AC-00099 DURATION: 10 TICKS

EVENT NAME: "scenario-1-event-3b"
HOSTILE TRACKS KILLED: 4
HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
FRIENDLY FIGHTERS USED: 8
NUMBER OF SCALE EXPANSION CHANGES: 0
NUMBER OF OFFSETS DONE: 0
Data For Killed Tracks, at time of Kill:

Total Time Traveled	Total Distance Traveled	Distance Above ADIZ
115.00Sec	014.37NM	045.99NM
123.50Sec	015.44NM	053.02NM
298.00Sec	035.18NM	075.95NM
234.50Sec	027.68NM	049.89NM

Communication-data
CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
TO: TACC DURATION: 10 TICKS
TO: TACC DURATION: 10 TICKS
TO: WAO DURATION: 10 TICKS
CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 21)
TO: SD DURATION: 10 TICKS
TO: AIRBASE-00115 DURATION: 10 TICKS
TO: AIRBASE-00115 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
TO: AIRBASE-00115 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS

```
        TO: AIRBASE-00115  DURATION: 10 TICKS
        TO: AIRBASE-00115  DURATION: 10 TICKS
        TO: WD2  DURATION: 10 TICKS
        TO: WD2  DURATION: 10 TICKS
        TO: WD2  DURATION: 10 TICKS
        TO: WD2  DURATION: 10 TICKS
        TO: AIRBASE-00115  DURATION: 10 TICKS
        TO: AIRBASE-00115  DURATION: 10 TICKS
        TO: WD2  DURATION: 10 TICKS
        TO: WD2  DURATION: 10 TICKS
        TO: WD2  DURATION: 10 TICKS
        TO: WD2  DURATION: 10 TICKS
        TO: WD2  DURATION: 10 TICKS
        TO: WD2  DURATION: 10 TICKS
    CREWMEMBER: WD2 (TOTAL NUMBER COMMUNS 17)
        TO: AC-00111  DURATION: 10 TICKS
        TO: AC-00110  DURATION: 10 TICKS
        TO: AC-00110  DURATION: 10 TICKS
        TO: AC-00111  DURATION: 10 TICKS
        TO: AC-00110  DURATION: 10 TICKS
        TO: AC-00110  DURATION: 10 TICKS
        TO: AC-00110  DURATION: 10 TICKS
        TO: AC-00110  DURATION: 10 TICKS
        TO: AC-00110  DURATION: 10 TICKS
        TO: AC-00121  DURATION: 10 TICKS
        TO: AC-00111  DURATION: 10 TICKS
        TO: AC-00121  DURATION: 10 TICKS
        TO: AC-00111  DURATION: 10 TICKS
        TO: AC-00111  DURATION: 10 TICKS
        TO: AC-00111  DURATION: 10 TICKS
        TO: AC-00111  DURATION: 10 TICKS
        TO: AC-00111  DURATION: 10 TICKS


EVENT NAME:  "scenario-2-event-1a"
                         HOSTILE TRACKS KILLED: 1
HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
                        FRIENDLY FIGHTERS USED: 6
       NUMBER OF SCALE EXPANSION CHANGES: 0
                       NUMBER OF OFFSETS DONE: 0
Data For Killed Tracks, at time of Kill:
 Total Time Traveled   Total Distance Traveled   Distance Above ADIZ
      145.00Sec              018.13NM                   053.81NM
Communication-data
   CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
       TO: TACC  DURATION: 10 TICKS
       TO: TACC  DURATION: 10 TICKS
       TO: WAO  DURATION: 10 TICKS
   CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 12)
       TO: SD  DURATION: 10 TICKS
       TO: WD1  DURATION: 10 TICKS
       TO: WD2  DURATION: 10 TICKS
       TO: WD1  DURATION: 10 TICKS
       TO: AIRBASE-00130  DURATION: 10 TICKS
       TO: AIRBASE-00130  DURATION: 10 TICKS
       TO: WD1  DURATION: 10 TICKS
       TO: WD1  DURATION: 10 TICKS
       TO: WD1  DURATION: 10 TICKS
       TO: WD1  DURATION: 10 TICKS
       TO: WD1  DURATION: 10 TICKS
       TO: WD1  DURATION: 10 TICKS
   CREWMEMBER: WD1 (TOTAL NUMBER COMMUNS 8)
       TO: AC-00128  DURATION: 10 TICKS
       TO: AC-00128  DURATION: 10 TICKS
       TO: AC-00128  DURATION: 10 TICKS
       TO: AC-00128  DURATION: 10 TICKS
       TO: AC-00128  DURATION: 10 TICKS
       TO: AC-00128  DURATION: 10 TICKS
       TO: AC-00128  DURATION: 10 TICKS
       TO: AC-00128  DURATION: 10 TICKS


EVENT NAME:  "scenario-2-event-1b"
                         HOSTILE TRACKS KILLED: 1
```

```
HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
              FRIENDLY FIGHTERS USED: 6
     NUMBER OF SCALE EXPANSION CHANGES: 0
              NUMBER OF OFFSETS DONE: 0
Data For Killed Tracks, at time of Kill:
 Total Time Traveled   Total Distance Traveled   Distance Above ADIZ
       145.00Sec              018.13NM                   053.81NM
Communication-data
  CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
     TO: TACC  DURATION: 10 TICKS
     TO: TACC  DURATION: 10 TICKS
     TO: WAO  DURATION: 10 TICKS
  CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 12)
     TO: SD  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: AIRBASE-00154  DURATION: 10 TICKS
     TO: AIRBASE-00154  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
  CREWMEMBER: WD2 (TOTAL NUMBER COMMUNS 4)
     TO: AC-00149  DURATION: 10 TICKS
     TO: AC-00149  DURATION: 10 TICKS
     TO: AC-00149  DURATION: 10 TICKS
     TO: AC-00149  DURATION: 10 TICKS


EVENT NAME:  "scenario-2-event-2a"
              HOSTILE TRACKS KILLED: 2
HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
              FRIENDLY FIGHTERS USED: 5
     NUMBER OF SCALE EXPANSION CHANGES: 0
              NUMBER OF OFFSETS DONE: 0
Data For Killed Tracks, at time of Kill:
 Total Time Traveled   Total Distance Traveled   Distance Above ADIZ
       094.00Sec              011.75NM                   055.26NM
       142.00Sec              017.75NM                   058.20NM
Communication-data
  CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
     TO: TACC  DURATION: 10 TICKS
     TO: TACC  DURATION: 10 TICKS
     TO: WAO  DURATION: 10 TICKS
  CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 8)
     TO: SD  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: AIRBASE-00171  DURATION: 10 TICKS
     TO: AIRBASE-00171  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
  CREWMEMBER: WD1 (TOTAL NUMBER COMMUNS 9)
     TO: AC-00166  DURATION: 10 TICKS
     TO: AC-00166  DURATION: 10 TICKS
     TO: AC-00167  DURATION: 10 TICKS
     TO: AC-00166  DURATION: 10 TICKS
     TO: AC-00167  DURATION: 10 TICKS
     TO: AC-00167  DURATION: 10 TICKS
     TO: AC-00167  DURATION: 10 TICKS
     TO: AC-00166  DURATION: 10 TICKS
     TO: AC-00166  DURATION: 10 TICKS


EVENT NAME:  "scenario-2-event-2b"
              HOSTILE TRACKS KILLED: 2
HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
              FRIENDLY FIGHTERS USED: 5
     NUMBER OF SCALE EXPANSION CHANGES: 0
              NUMBER OF OFFSETS DONE: 0
```

Data For Killed Tracks, at time of Kill:

Total Time Traveled	Total Distance Traveled	Distance Above ADIZ
083.00Sec	010.38NM	052.32NM
140.50Sec	017.56NM	057.80NM

Communication-data

CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
TO: TACC DURATION: 10 TICKS
TO: TACC DURATION: 10 TICKS
TO: WAO DURATION: 10 TICKS
CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 8)
TO: SD DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
TO: AIRBASE-00181 DURATION: 10 TICKS
TO: AIRBASE-00181 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
CREWMEMBER: WD2 (TOTAL NUMBER COMMUNS 7)
TO: AC-00179 DURATION: 10 TICKS
TO: AC-00180 DURATION: 10 TICKS
TO: AC-00180 DURATION: 10 TICKS
TO: AC-00179 DURATION: 10 TICKS
TO: AC-00180 DURATION: 10 TICKS
TO: AC-00179 DURATION: 10 TICKS
TO: AC-00179 DURATION: 10 TICKS

EVENT NAME: "scenario-2-event-3a"
HOSTILE TRACKS KILLED: 4
HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
FRIENDLY FIGHTERS USED: 6
NUMBER OF SCALE EXPANSION CHANGES: 0
NUMBER OF OFFSETS DONE: 0
Data For Killed Tracks, at time of Kill:

Total Time Traveled	Total Distance Traveled	Distance Above ADIZ
105.00Sec	012.40NM	038.30NM
151.00Sec	017.83NM	049.37NM
258.50Sec	030.52NM	064.67NM
281.00Sec	033.17NM	070.08NM

Communication-data

CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
TO: TACC DURATION: 10 TICKS
TO: TACC DURATION: 10 TICKS
TO: WAO DURATION: 10 TICKS
CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 13)
TO: SD DURATION: 10 TICKS
TO: AIRBASE-00191 DURATION: 10 TICKS
TO: AIRBASE-00191 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
TO: AIRBASE-00191 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: AIRBASE-00191 DURATION: 10 TICKS
TO: AIRBASE-00191 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
CREWMEMBER: WD1 (TOTAL NUMBER COMMUNS 14)
TO: AC-00186 DURATION: 10 TICKS
TO: AC-00186 DURATION: 10 TICKS
TO: AC-00187 DURATION: 10 TICKS
TO: AC-00186 DURATION: 10 TICKS
TO: AC-00187 DURATION: 10 TICKS
TO: AC-00186 DURATION: 10 TICKS
TO: AC-00187 DURATION: 10 TICKS
TO: AC-00187 DURATION: 10 TICKS
TO: AC-00186 DURATION: 10 TICKS
TO: AC-00187 DURATION: 10 TICKS
TO: AC-00186 DURATION: 10 TICKS
TO: AC-00186 DURATION: 10 TICKS
TO: AC-00187 DURATION: 10 TICKS
TO: AC-00187 DURATION: 10 TICKS

EVENT NAME: "scenario-2-event-3b"
HOSTILE TRACKS KILLED: 4
HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
FRIENDLY FIGHTERS USED: 6
NUMBER OF SCALE EXPANSION CHANGES: 0
NUMBER OF OFFSETS DONE: 0
Data For Killed Tracks, at time of Kill:

Total Time Traveled	Total Distance Traveled	Distance Above ADIZ
336.00Sec	039.67NM	083.20NM
133.50Sec	015.76NM	045.16NM
098.00Sec	011.57NM	036.61NM
232.50Sec	027.45NM	058.41NM

Communication-data
CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
TO: TACC DURATION: 10 TICKS
TO: TACC DURATION: 10 TICKS
TO: WAO DURATION: 10 TICKS
CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 11)
TO: SD DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
TO: AIRBASE-00204 DURATION: 10 TICKS
TO: AIRBASE-00204 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
TO: AIRBASE-00204 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
CREWMEMBER: WD2 (TOTAL NUMBER COMMUNS 15)
TO: AC-00202 DURATION: 10 TICKS
TO: AC-00203 DURATION: 10 TICKS
TO: AC-00203 DURATION: 10 TICKS
TO: AC-00202 DURATION: 10 TICKS
TO: AC-00203 DURATION: 10 TICKS
TO: AC-00202 DURATION: 10 TICKS
TO: AC-00203 DURATION: 10 TICKS
TO: AC-00203 DURATION: 10 TICKS
TO: AC-00209 DURATION: 10 TICKS
TO: AC-00209 DURATION: 10 TICKS
TO: AC-00202 DURATION: 10 TICKS
TO: AC-00209 DURATION: 10 TICKS
TO: AC-00202 DURATION: 10 TICKS
TO: AC-00209 DURATION: 10 TICKS
TO: AC-00209 DURATION: 10 TICKS

EVENT NAME: "scenario-3-event-3b"
HOSTILE TRACKS KILLED: 1
HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
FRIENDLY FIGHTERS USED: 8
NUMBER OF SCALE EXPANSION CHANGES: 0
NUMBER OF OFFSETS DONE: 0
Data For Killed Tracks, at time of Kill:

Total Time Traveled	Total Distance Traveled	Distance Above ADIZ
254.50Sec	030.05NM	057.28NM
272.50Sec	032.17NM	072.61NM
153.50Sec	018.12NM	053.96NM
112.50Sec	013.28NM	054.69NM

Communication-data
CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
TO: TACC DURATION: 10 TICKS
TO: TACC DURATION: 10 TICKS
TO: WAO DURATION: 10 TICKS
CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 14)
TO: SD DURATION: 10 TICKS
TO: AIRBASE-00217 DURATION: 10 TICKS
TO: AIRBASE-00217 DURATION: 10 TICKS
TO: WD1 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
TO: AIRBASE-00217 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
TO: AIRBASE-00217 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS
TO: WD2 DURATION: 10 TICKS

```
      TO: WD2  DURATION: 10 TICKS
      TO: WD2  DURATION: 10 TICKS
      TO: WD2  DURATION: 10 TICKS
      TO: WD2  DURATION: 10 TICKS
   CREWMEMBER: WD2 (TOTAL NUMBER COMMUNS 11)
      TO: AC-00215  DURATION: 10 TICKS
      TO: AC-00216  DURATION: 10 TICKS
      TO: AC-00215  DURATION: 10 TICKS
      TO: AC-00216  DURATION: 10 TICKS
      TO: AC-00216  DURATION: 10 TICKS
      TO: AC-00216  DURATION: 10 TICKS
      TO: AC-00215  DURATION: 10 TICKS
      TO: AC-00216  DURATION: 10 TICKS
      TO: AC-00215  DURATION: 10 TICKS
      TO: AC-00216  DURATION: 10 TICKS
      TO: AC-00216  DURATION: 10 TICKS


EVENT NAME:  "scenario-3-event-3a"
                  HOSTILE TRACKS KILLED: 4
HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
                FRIENDLY FIGHTERS USED: 8
    NUMBER OF SCALE EXPANSION CHANGES: 0
                NUMBER OF OFFSETS DONE: 0
Data For Killed Tracks, at time of Kill:
 Total Time Traveled   Total Distance Traveled   Distance Above ADIZ
        228.50Sec              026.98NM                 050.85NM
        202.00Sec              023.85NM                 055.16NM
        151.00Sec              017.83NM                 053.34NM
        099.50Sec              011.75NM                 051.47NM
Communication-data
  CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
     TO: TACC  DURATION: 10 TICKS
     TO: TACC  DURATION: 10 TICKS
     TO: WAO  DURATION: 10 TICKS
  CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 22)
     TO: SD  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: AIRBASE-00232  DURATION: 10 TICKS
     TO: AIRBASE-00232  DURATION: 10 TICKS
     TO: AIRBASE-00232  DURATION: 10 TICKS
     TO: AIRBASE-00232  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: AIRBASE-00232  DURATION: 10 TICKS
     TO: AIRBASE-00232  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
  CREWMEMBER: WD1 (TOTAL NUMBER COMMUNS 12)
     TO: AC-00227  DURATION: 10 TICKS
     TO: AC-00227  DURATION: 10 TICKS
     TO: AC-00228  DURATION: 10 TICKS
     TO: AC-00227  DURATION: 10 TICKS
     TO: AC-00228  DURATION: 10 TICKS
     TO: AC-00227  DURATION: 10 TICKS
     TO: AC-00227  DURATION: 10 TICKS
     TO: AC-00227  DURATION: 10 TICKS
     TO: AC-00228  DURATION: 10 TICKS
     TO: AC-00228  DURATION: 10 TICKS
     TO: AC-00227  DURATION: 10 TICKS
     TO: AC-00231  DURATION: 10 TICKS
  CREWMEMBER: WD2 (TOTAL NUMBER COMMUNS 2)
     TO: AC-00231  DURATION: 10 TICKS
     TO: AC-00231  DURATION: 10 TICKS
```

```
EVENT NAME:  "scenario-3-event-2b"
                    HOSTILE TRACKS KILLED: 2
HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
                  FRIENDLY FIGHTERS USED: 5
    NUMBER OF SCALE EXPANSION CHANGES: 0
                  NUMBER OF OFFSETS DONE: 0
Data For Killed Tracks, at time of Kill:
 Total Time Traveled   Total Distance Traveled   Distance Above ADIZ
       122.00Sec                014.40NM                    053.86NM
       119.50Sec                014.11NM                    048.65NM
Communication-data
  CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
     TO: TACC  DURATION: 10 TICKS
     TO: TACC  DURATION: 10 TICKS
     TO: WAO  DURATION: 10 TICKS
  CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 8)
     TO: SD  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: AIRBASE-00247  DURATION: 10 TICKS
     TO: AIRBASE-00247  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
  CREWMEMBER: WD2 (TOTAL NUMBER COMMUNS 8)
     TO: AC-00246  DURATION: 10 TICKS
     TO: AC-00245  DURATION: 10 TICKS
     TO: AC-00245  DURATION: 10 TICKS
     TO: AC-00246  DURATION: 10 TICKS
     TO: AC-00245  DURATION: 10 TICKS
     TO: AC-00246  DURATION: 10 TICKS
     TO: AC-00246  DURATION: 10 TICKS
     TO: AC-00245  DURATION: 10 TICKS


EVENT NAME:  "scenario-3-event-2a"
                    HOSTILE TRACKS KILLED: 2
HOSTILE TRACKS KILLED OUTSIDE OF ADIZ: 0
                  FRIENDLY FIGHTERS USED: 5
    NUMBER OF SCALE EXPANSION CHANGES: 0
                  NUMBER OF OFFSETS DONE: 0
Data For Killed Tracks, at time of Kill:
 Total Time Traveled   Total Distance Traveled   Distance Above ADIZ
       124.00Sec                014.64NM                    054.35NM
       145.50Sec                017.18NM                    054.86NM
Communication-data
  CREWMEMBER: SD (TOTAL NUMBER COMMUNS 3)
     TO: TACC  DURATION: 10 TICKS
     TO: TACC  DURATION: 10 TICKS
     TO: WAO  DURATION: 10 TICKS
  CREWMEMBER: WAO (TOTAL NUMBER COMMUNS 9)
     TO: SD  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD2  DURATION: 10 TICKS
     TO: AIRBASE-00257  DURATION: 10 TICKS
     TO: AIRBASE-00257  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: WD1  DURATION: 10 TICKS
     TO: AIRBASE-00237  DURATION: 10 TICKS
  CREWMEMBER: WD1 (TOTAL NUMBER COMMUNS 10)
     TO: AC-00252  DURATION: 10 TICKS
     TO: AC-00252  DURATION: 10 TICKS
     TO: AC-00253  DURATION: 10 TICKS
     TO: AC-00252  DURATION: 10 TICKS
     TO: AC-00253  DURATION: 10 TICKS
     TO: AC-00252  DURATION: 10 TICKS
     TO: AC-00252  DURATION: 10 TICKS
     TO: AC-00253  DURATION: 10 TICKS
     TO: AC-00253  DURATION: 10 TICKS
     TO: AC-00252  DURATION: 10 TICKS
```